LUCA NOVELLI

MI PRIMER LIBRO DE BASIC

EL LENGUAJE MÁS DIFUNDIDO DE LOS ORDENADORES

ESTE LIBRO ES CASI UN DICCIONARIO.

EN CADA "PALABRA" HAY UN EJERCICIO DIVERTIDO

SE APRENDE FÁCILMENTE

OS VALE TAMBIÉN EN EL "COLE"

ES INTERNACIONAL

Y, ADEMÁS, TAMBIÉN ESTOY YO...



mi primer libro de

BASIC

LUCA NOVELLI

Título original: *Il mio primo libro di Basic,* Milán, 1984
Traducción: José Golacheca

1.ª edición, septiembre 1984

Villafranca, 22. 28028 Madrid
ISBN: 84-7525-175-7
Depósito legal: M. 33.627 - 1984
Imprime: Litografía Josmar, S. A.
Polígono Industrial de Coslada (Madrid)
Printed in Spain

LUCA NOVELLI

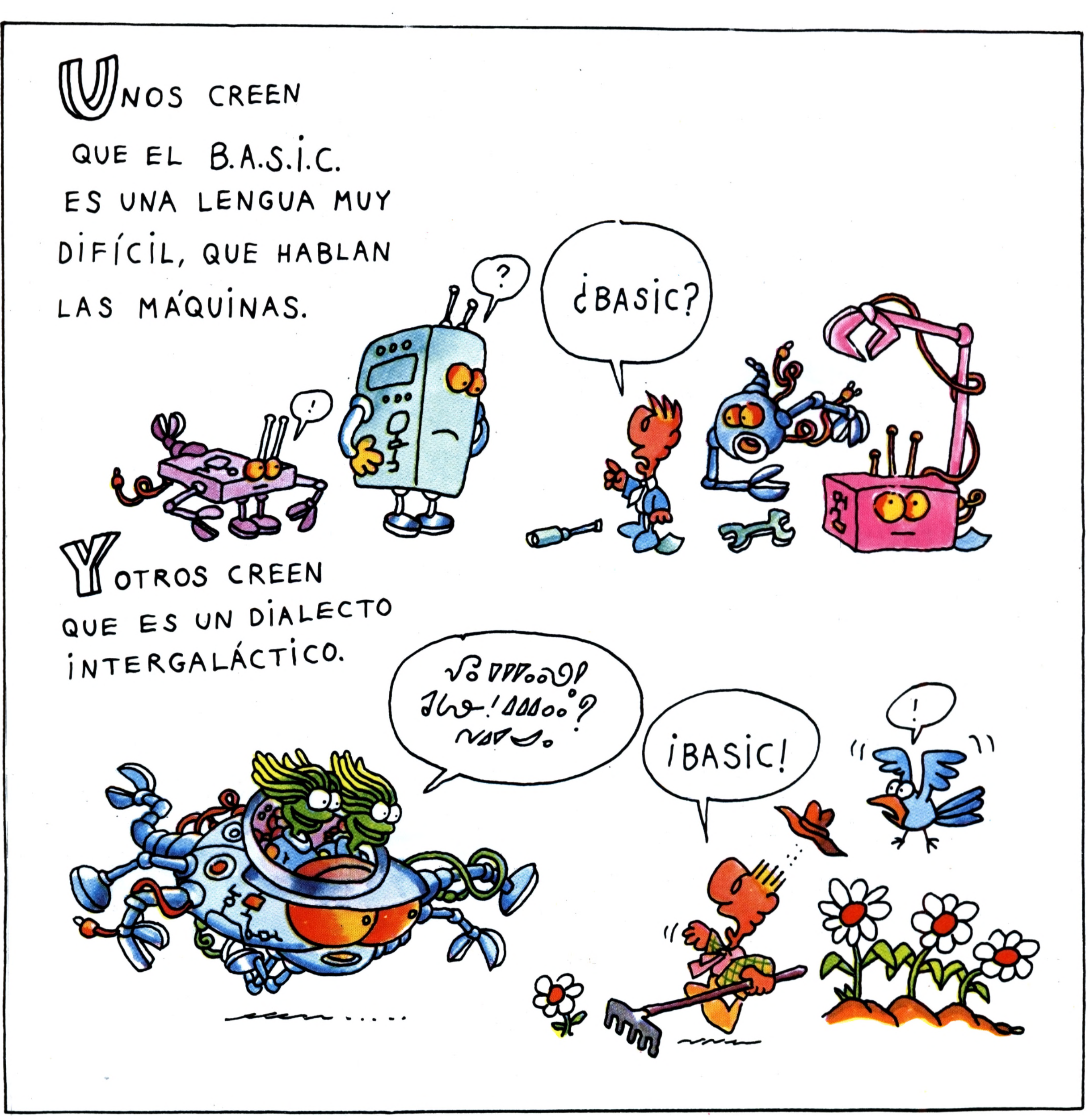
MI PRIMER LIBRO DE



EDICIONES GENERALES ANAYA

Punto de partida

Donde se descubre
que el BASIC
no es una lengua
extraterrestre

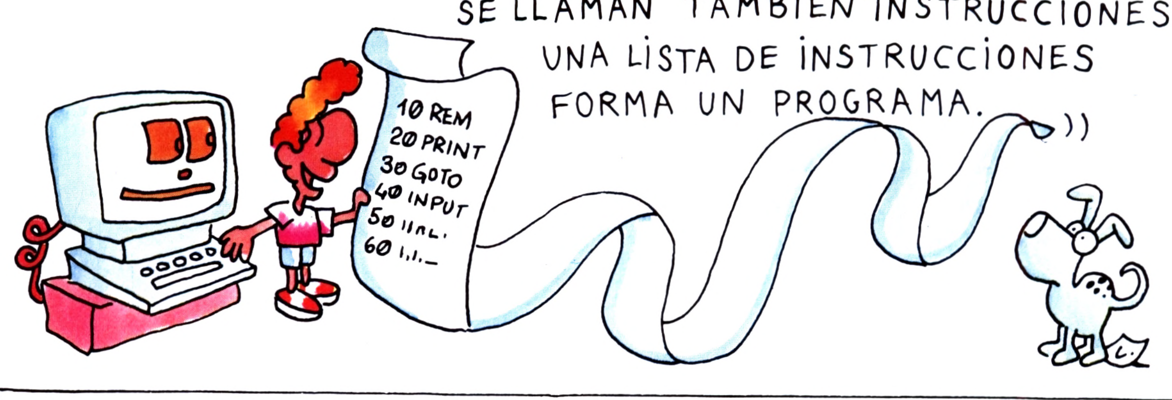


SIN EMBARGO EL B.A.S.I.C. ES EL LENGUAJE MÁS DIFUNDIDO DE LOS ORDENADORES.

La sigla BASIC es la abreviatura de Beginners' All-purpose Symbolic Instruction Code (Código de instrucciones de uso universal para principiantes). Comprende un vocabulario de más de 200 términos que, en su mayoría, son muy corrientes en la lengua inglesa, tales como AND (= y), PRINT (= imprime), GOTO (= vete a), LEFT (= izquierda), RIGHT (= derecha) y otros muchos.



LAS "ÓRDENES" QUE DAIS AL ORDENADOR SE LLAMAN TAMBIÉN "INSTRUCCIONES". UNA LISTA DE INSTRUCCIONES FORMA UN PROGRAMA.



PROGRAMAS MUY CORTOS PUEDEN DESARROLLAR "TRABAJOS" MUY LARGOS Y COMPLICADOS.

UN VIDEOJUEGO Y EL PLAN DE VUELO DEL SPACE SHUTTLE SON "PROGRAMAS".

CON ESTE LIBRO APRENDERÉIS LOS MÁS IMPORTANTES VOCABLOS BASIC Y PODRÉIS HACER "DESARROLLAR" A VUESTRO ORDENADOR PROGRAMAS BASIC.

```
10 REM
20 PRINT
30 INPUT A$
40 ...
```

CUANDO HAYÁIS TRANSCRITO UN PROGRAMA, LO PODRÉIS CONSERVAR Y UTILIZAR CUANTAS VECES QUERÁIS.



¡CUIDADO CON EQUIVOCAROS!

¡EL ORDENADOR ES PEDANTE Y PESADO!
SI EQUIVOCÁIS UN VOCABLO U OLVIDÁIS UNA COMA, EL ORDENADOR NO OBEDECERÁ VUESTRAS ÓRDENES.
Y EN EL MEJOR DE LOS CASOS INMEDIATAMENTE OS SEÑALARÁ EL ERROR "DE SINTAXIS".



UN CONSEJO:
ES INÚTIL QUE OS ENFADÉIS CON UN ORDENADOR.



ES UNA MÁQUINA QUE HARÁ SIEMPRE Y SÓLO LO QUE DIGÁIS QUE HAGA, EN BASIC, NATURALMENTE.

Gramática Basic

Donde descubrimos que el ordenador obedece también al punto y coma



CON EL PROGRAMA, JUNTO CON INSTRUCCIONES, DAD AL ORDENADOR TAMBIÉN PALABRAS Y NÚMEROS.

CADENA. Es una secuencia de caracteres (palabras, letras del alfabeto, símbolos gráficos y espacios vacíos) puesta entre comillas. "¡AUGH!" es una cadena de seis caracteres.

" " LAS COMILLAS. El ordenador reconoce con ellas el principio y el final de una cadena que hay que imprimir o tratar de algún otro modo.

CONSTANTES. Son los caracteres entre comillas y los valores numéricos que permanecen invariables en el programa.





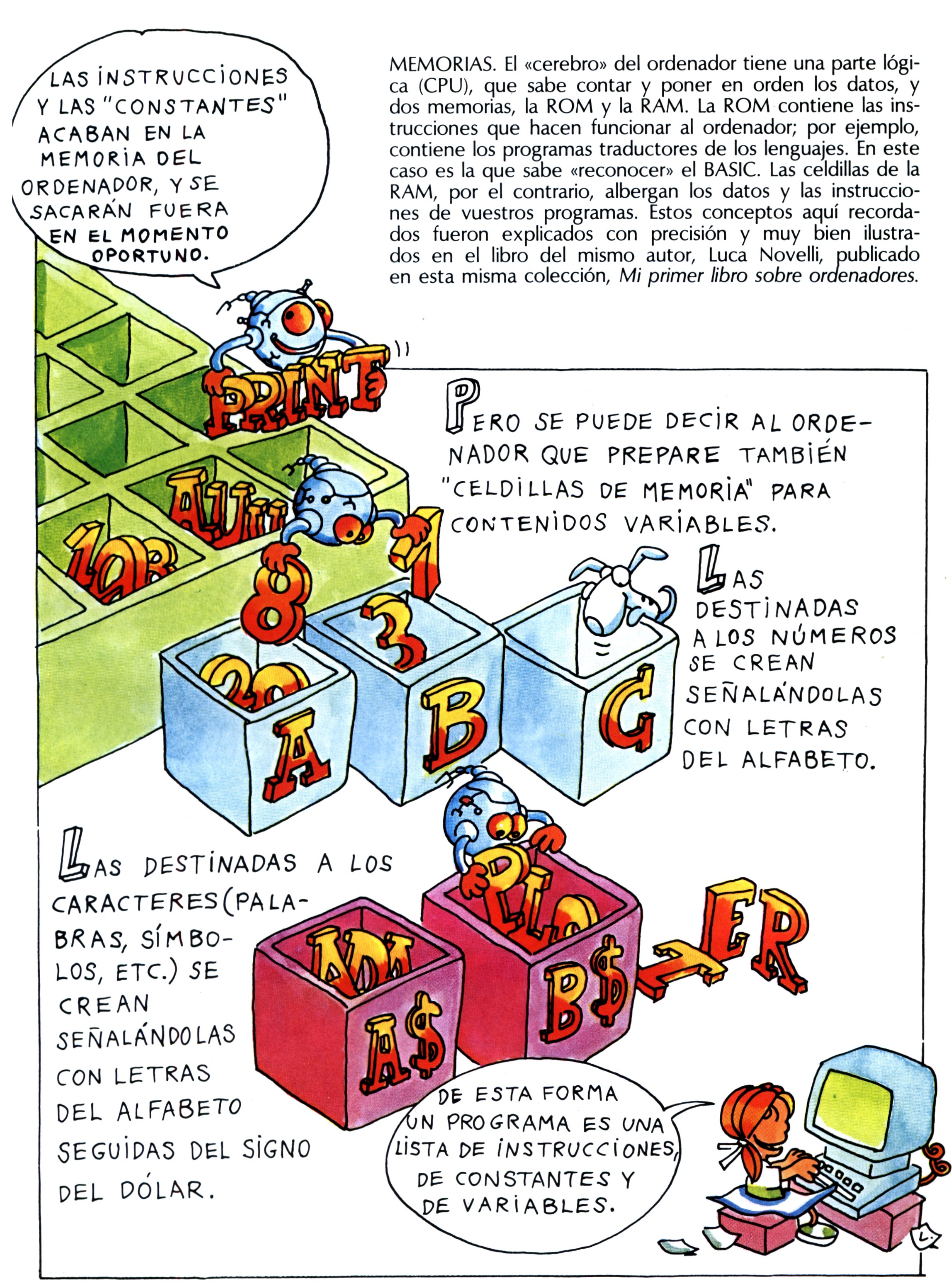
; PUNTO Y COMA. Manda al ordenador que imprima a continuación también el contenido de la siguiente instrucción PRINT.

, COMA. Manda al ordenador que ponga en columna los datos que se van a imprimir. Sirve también para dividir las variables de entrada.

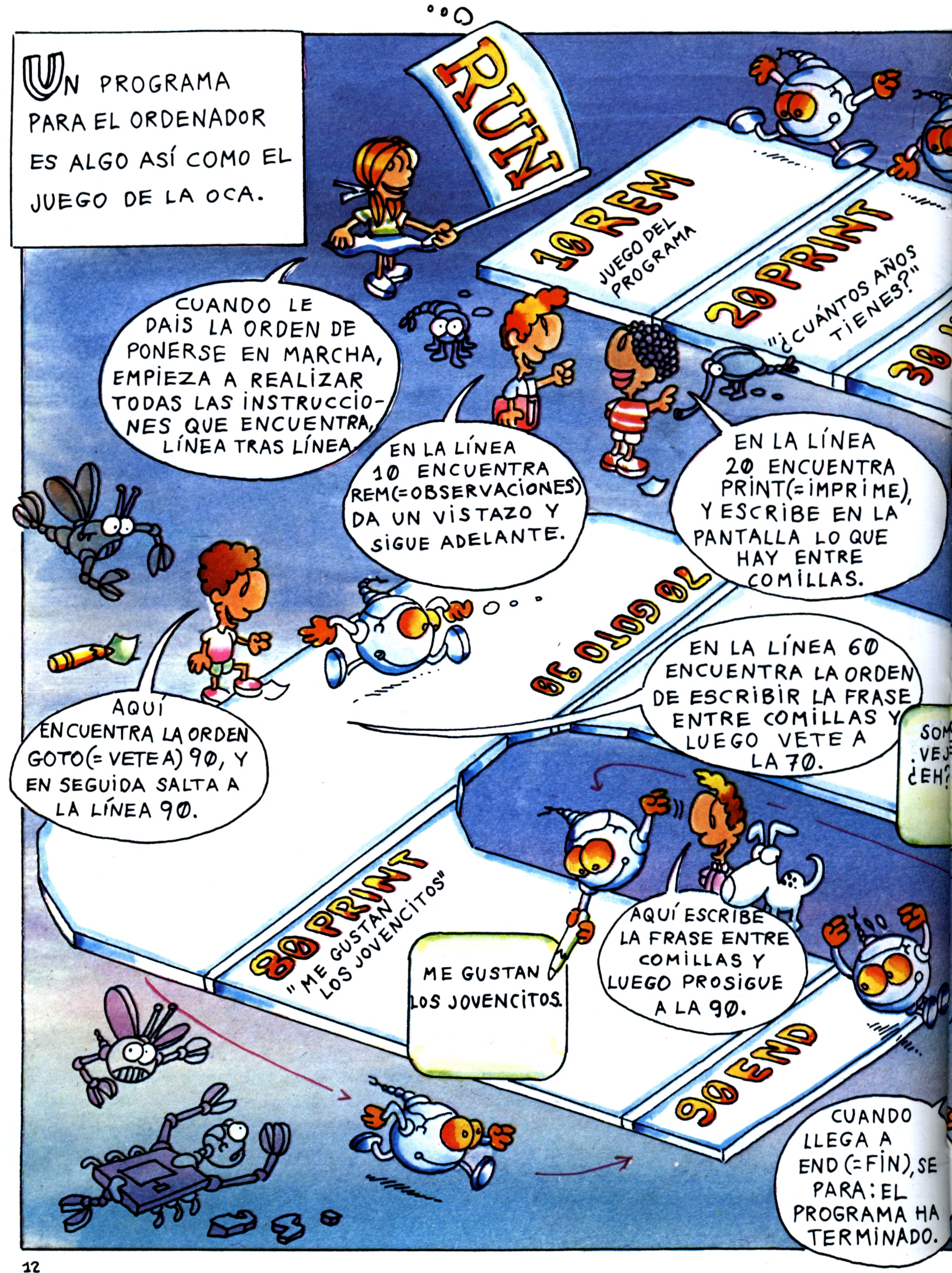
: DOS PUNTOS. Si al final de una instrucción hay dos puntos, se puede añadir otra instrucción en la misma línea.

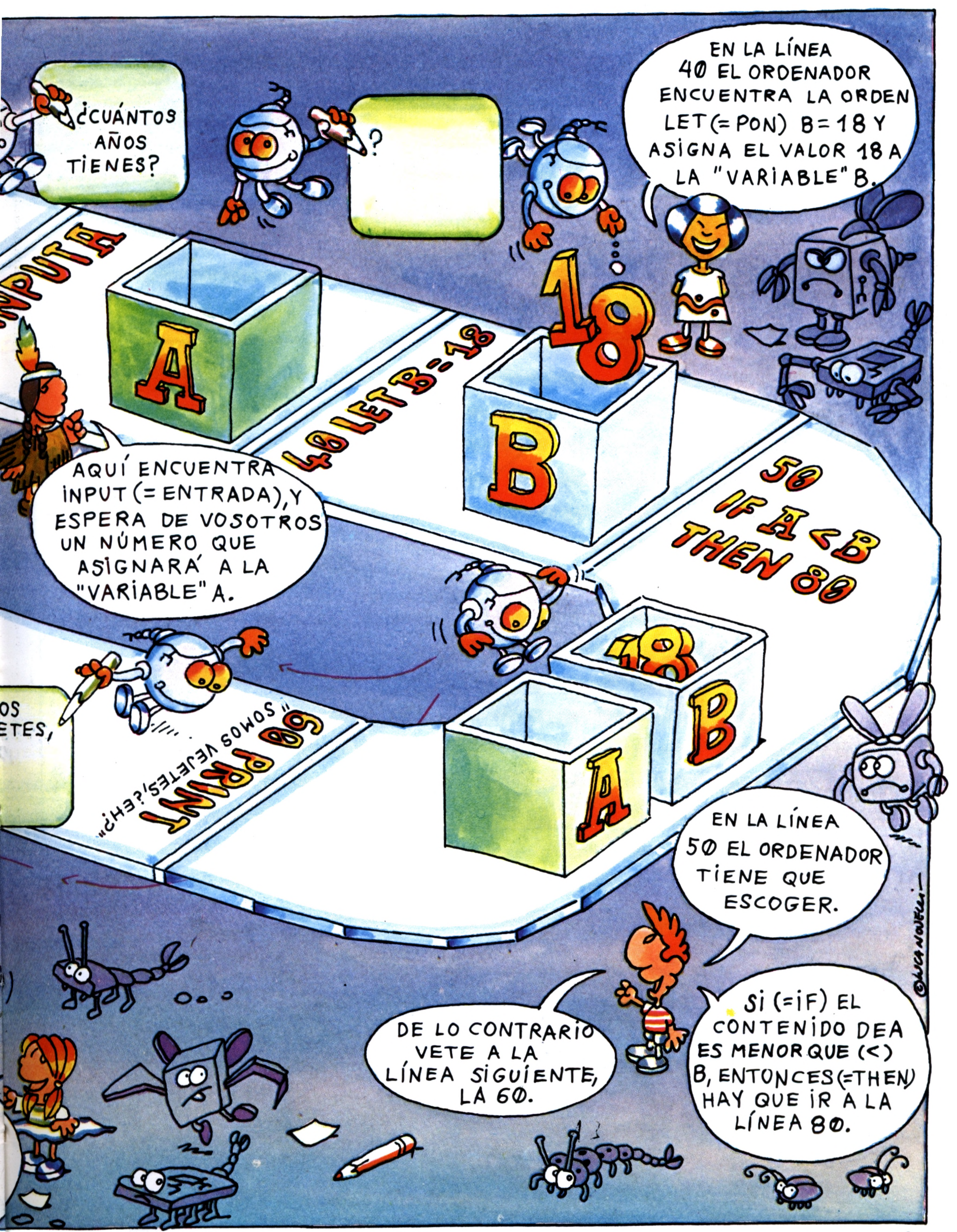
. PUNTO. Divide las unidades de los decimales. Por ejemplo, el número 167,61, el ordenador lo escribe 167.61.

MEMORIAS. El «cerebro» del ordenador tiene una parte lógica (CPU), que sabe contar y poner en orden los datos, y dos memorias, la ROM y la RAM. La ROM contiene las instrucciones que hacen funcionar al ordenador; por ejemplo, contiene los programas traductores de los lenguajes. En este caso es la que sabe «reconocer» el BASIC. Las celdillas de la RAM, por el contrario, albergan los datos y las instrucciones de vuestros programas. Estos conceptos aquí recordados fueron explicados con precisión y muy bien ilustrados en el libro del mismo autor, Luca Novelli, publicado en esta misma colección, *Mi primer libro sobre ordenadores*.



UN PROGRAMA PARA EL ORDENADOR ES ALGO ASÍ COMO EL JUEGO DE LA OCA.





¿CUÁNTOS AÑOS TIENES?
EN LA LÍNEA 40 EL ORDENADOR ENCUENTRA LA ORDEN LET (= PON) B = 18 Y ASIGNA EL VALOR 18 A LA "VARIABLE" B.
INPUT A
18
A
40 LET B = 18
B
AQUÍ ENCUENTRA INPUT (= ENTRADA), Y ESPERA DE VOSOTROS UN NÚMERO QUE ASIGNARÁ A LA "VARIABLE" A.
50 IF A < B THEN 80
18
B
A
60 PRINT "SOMOS VEJETES, ¿EH?"
EN LA LÍNEA 50 EL ORDENADOR TIENE QUE ESCOGER.
SI (= IF) EL CONTENIDO DE A ES MENOR QUE (<) B, ENTONCES (= THEN) HAY QUE IR A LA LÍNEA 80.
DE LO CONTRARIO VETE A LA LÍNEA SIGUIENTE, LA 60.



DIAGRAMA DE FLUJO. Lo llaman también «flow chart» o diagrama en bloques. Es la forma gráfica más sencilla para describir el esquema de un programa. Cada bloque da lugar a una línea de instrucciones. Si el ordenador tiene que tomar una decisión, los bloques se representan por un rombo. Las instrucciones GOTO (=vete a) se representan con una flecha.

10 START

20 PRINT "¿CUÁNTOS AÑOS TIENES?"

30 INPUT A

40 LET B =18

50 IF A<B THEN

SI ES VERDAD

ENTONCES

NO ES VERDAD

60 PRINT "SOMOS VEJETES, ¿EH?"

80 PRINT "ME GUSTAN LOS JOVENCITOS"

70 GOTO

90 END

YO LO LLAMO DIAGRAMA DE FLUJO.

HACER UN DIAGRAMA DE FLUJO QUIERE DECIR: DIVIDIR UN PROBLEMA EN MUCHOS PASOS CORTOS, POR LO QUE HASTA UN CABEZÓN DE ORDENADOR PUEDE ENTENDER.

PERO BASTA YA CON LOS PREÁMBULOS Y VAMOS ALLÁ CON NUESTRO DICCIONARIO DE BASIC. EMPEZAMOS POR **AND.**

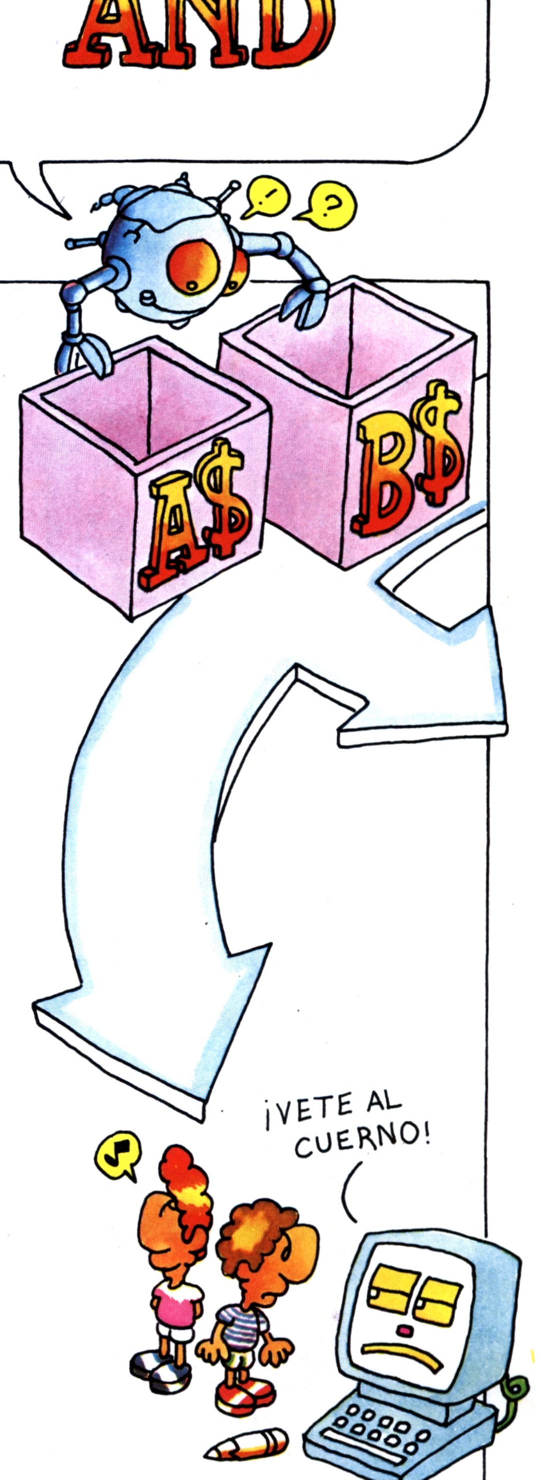
¡EH! ¡EH!

20 PRUNT

¡Y OJO A LAS ERRATAS!...

...QUE LOS EXPERTOS LLAMAN BUG (=BICHITOS)

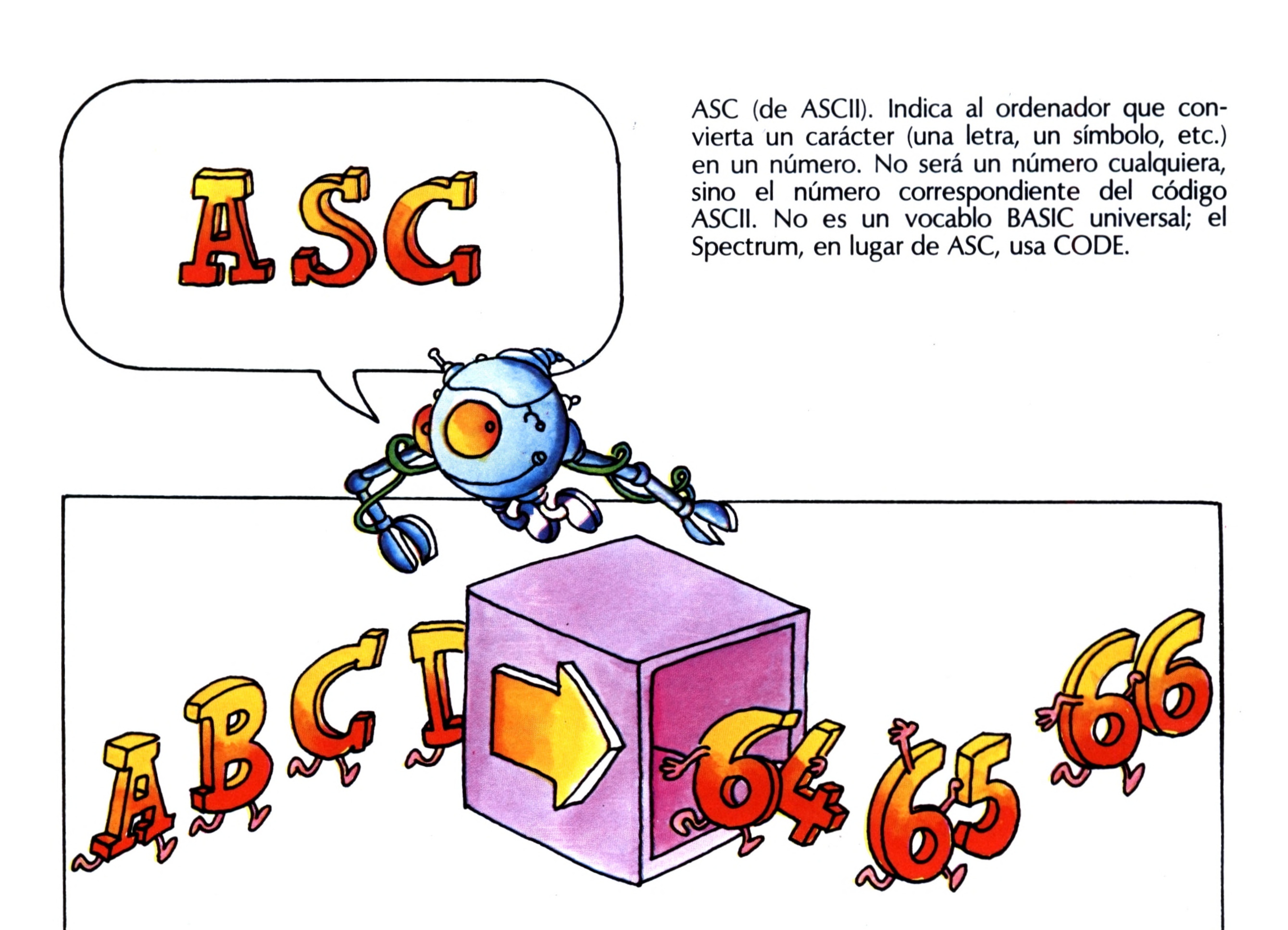
AND (= y). Es una «conjunción» también en BASIC. El ordenador obedece a la instrucción sólo cuando las condiciones puestas por A AND (y) B son verdaderas. En todos los demás casos el ordenador no hace nada y pasa a la línea siguiente.



PROGRAMITA DE BIENVENIDA

Muchos niños vienen a hacer una visita a Marko para ver y probar su ordenador. Por eso Marko ha escrito este pequeño programa para dar la bienvenida a sus amigos. Pero sólo si A$ = "SI" y B$ = "SI", el ordenador hará los honores de la casa, de lo contrario...

```
10 REM PROGRAMA DE BIENVENIDA
20 PRINT "HOLA,SOY EL ORDENADOR DE MARKO"
30 PRINT "TU QUIEN ERES?"
40 INPUT N$
50 PRINT "HOLA ";N$;" MUCHO GUSTO EN CONOCERTE"
60 PRINT "TE GUSTAN LOS ORDENADORES?"
70 PRINT "RESPONDE SI O NO"
80 INPUT A$
90 IF A$="SI" OR A$="NO" GOTO 110
100 GOTO 70
110 PRINT "QUIERES APRENDER BASIC?"
120 PRINT "RESPONDE SI O NO"
130 INPUT B$
140 IF B$="SI" OR B$="NO" GOTO 160
150 GOTO 120
160 IF A$="SI" AND B$="SI" GOTO 180
170 PRINT "VETE AL CUERNO! ";N$:END
180 PRINT "BIENVENIDO ENTRE NOSOTROS ";N$
```



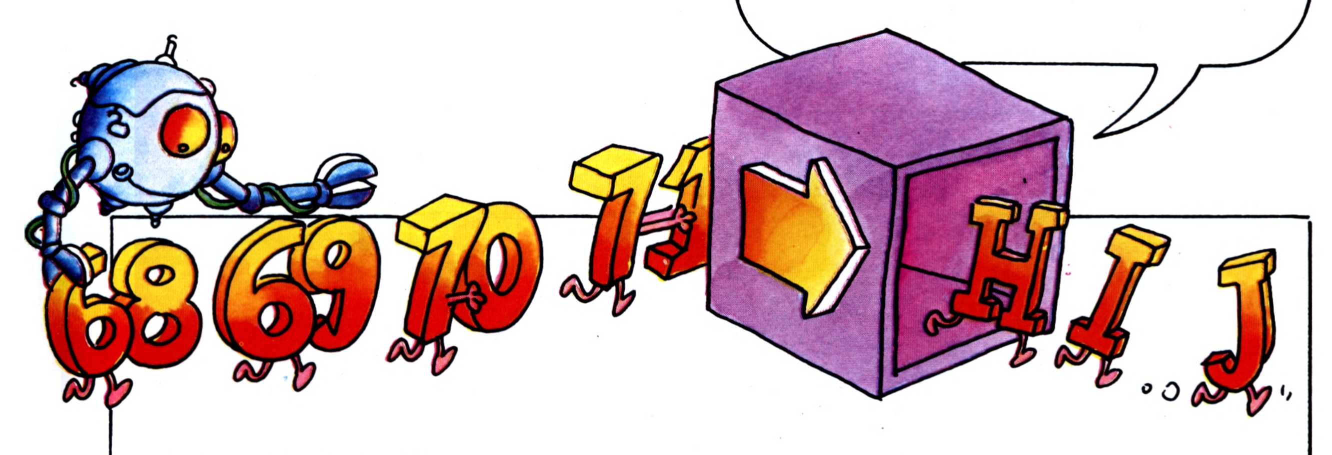
ASC (de ASCII). Indica al ordenador que convierta un carácter (una letra, un símbolo, etc.) en un número. No será un número cualquiera, sino el número correspondiente del código ASCII. No es un vocablo BASIC universal; el Spectrum, en lugar de ASC, usa CODE.

EN BUSCA DEL CODIGO ASCII

ASCII deriva de las iniciales de American Standard Code for Information Interchange (Código americano standard para el intercambio de informaciones). Ya es un código internacional que representa letras y números con secuencias de ocho bit (cero y uno). Las posibles secuencias de ocho bit son 256 (2^8) y, por tanto, pueden codificar 256 caracteres, símbolos gráficos, etc. Los números que descubriréis con nuestro ejercicio son la forma decimal de estas secuencias de números binarios. Por ejemplo, el número de código ASCII de la A es 65, y representa en forma decimal el número binario 01000001.

CHR$ (de character = carácter). Convierte números en caracteres (letras, símbolos, etc.). Esto es, realiza la operación inversa de ASC. Siguiendo en el ejemplo, PRINT CHR$ (65) visualizará la letra A, como está previsto en la tabla del código ASCII.






```
10 REM EL CODIGO ASCII
20 CLS:PRINT:PRINT
30 PRINT"PULSA :":PRINT
40 PRINT"     1    PARA VER LA TABLA ASCII":PRINT
50 PRINT"     2    PARA VER LA CORRESPONDENCIA"
60 PRINT"          CARACTER - CODIGO":PRINT
70 PRINT"     3    PARA VER LA CORRESPONDENCIA"
80 PRINT"          CODIGO - CARACTER":PRINT
90 PRINT"     4    PARA TERMINAR"
100 INPUT S
110 ON S GOTO 130,190,290,410
120 CLS
130 FOR I=1 TO 255
140 PRINT CHR$(I),I
150 NEXT I
160 PRINT"PULSA RETURN PARA CONTINUAR"
170 INPUT A$
180 IF A$=""GOTO 20
190 CLS:PRINT
200 PRINT"INTRODUCE EL CARACTER DEL QUE QUIERES VER EL CODIGO"
210 INPUT C$
220 PRINT:PRINT
230 PRINT"AL CARACTER ";C$;" CORRESPONDE EL CODIGO ASCII";ASC(C$)
240 PRINT:PRINT
250 PRINT"QUIERES VER OTRO ?"
260 INPUT A$
270 IF A$="SI" GOTO 190
280 GOTO 20
290 CLS:PRINT
300 PRINT"INTRODUCE EL CODIGO (COMPRENDIDO ENTRE 1 Y 255) DEL QUE"
310 PRINT"QUIERES VER EL CARACTER CORRESPONDIENTE"
320 INPUT C
330 IF C<1 OR C>255 GOTO 290
340 PRINT:PRINT
350 PRINT"AL CODIGO ";C;"CORRESPONDE EL CARACTER ";CHR$(C)
360 PRINT:PRINT
370 PRINT"QUIERES VER OTRO ?"
380 INPUT A$
390 IF A$="SI" GOTO 290
400 GOTO 20
410 CLS:END
```



CLS (de CLear the Screen = despeja la pantalla). Indica al ordenador que quite todo lo que hay en la pantalla. No es un vocablo BASIC universal; por ejemplo, Apple tiene HOME; VIC 20 y COMMODORE 64 tienen PRINT «{SHIFT CLR HOME}».

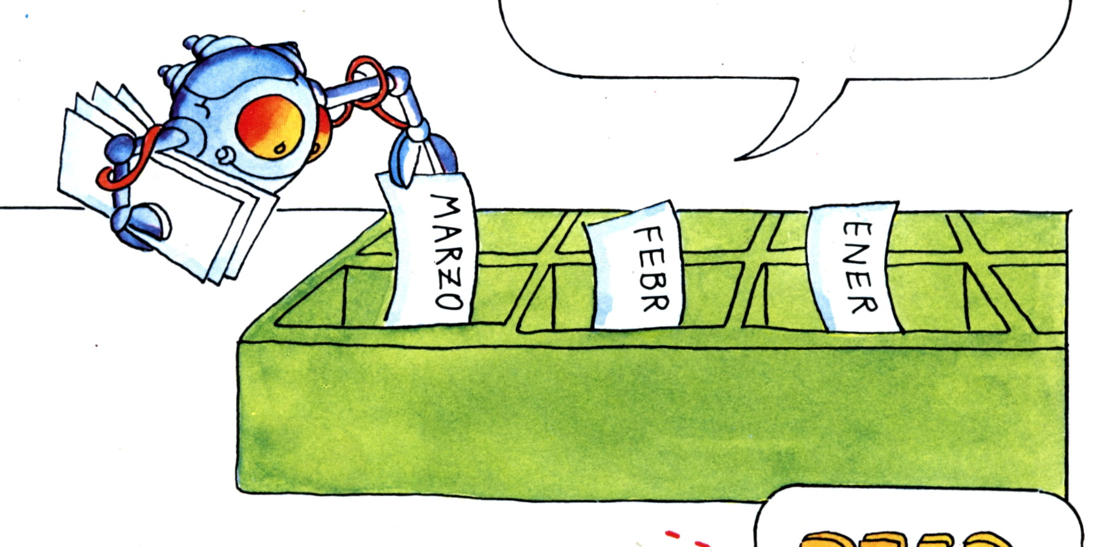
EL SECRETO DE LOS VIDEOJUEGOS

La orden CLS (o sus equivalentes) es el secreto de la animación en los videojuegos. El principio es muy sencillo: se da al ordenador la orden de imprimir (PRINT) un punto o un símbolo gráfico, inmediatamente después se le da la orden de borrarlo (CLS) y luego de reimprimirlo un poco más adelante. De esta forma, ante nuestros ojos se crea la ilusión óptica de que el punto se mueve en la pantalla. En realidad son muchos nuevos puntos que inmediatamente después se borran.



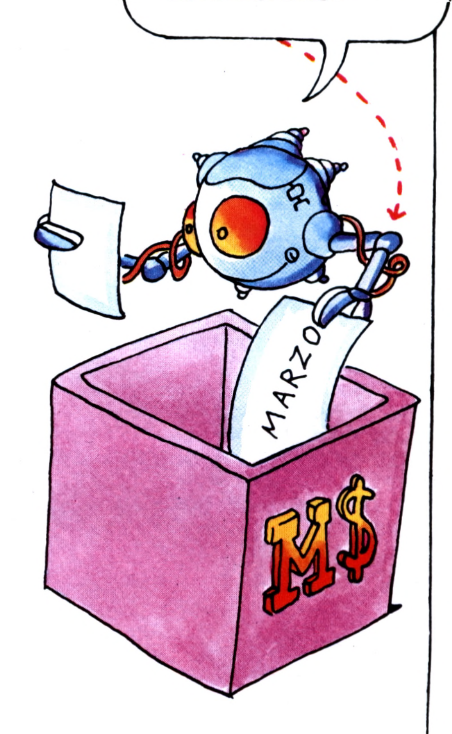
DATA (= datos). Prepara datos, números o cadenas. Estos datos luego se asociarán a las variables indicadas por la instrucción READ (= lee). Una tercera instrucción, RESTORE (= restablece), indicará al ordenador que reponga todo en las condiciones iniciales de DATA.





LOS MESES DEL AÑO

DATA se encuentra siempre junto con READ y RESTORE, formando una sola instrucción de asignación. En nuestro ejemplo, DATA «prepara» los meses del año, que luego se asocian por READ a la variable M$. La tercera instrucción, RESTORE, indicará al ordenador que reordene todo como estaba en la primera línea de DATA, permitiéndonos seguir.




```
10 DATA ENERO,FEBRERO,MARZO,ABRIL
20 DATA MAYO,JUNIO,JULIO,AGOSTO
30 DATA SEPTIEMBRE,OCTUBRE,NOVIEMBRE,DICIEMBRE
40 CLS:PRINT"DAME UN MES EN NUMERO"
50 INPUT M
60 M=INT (M)
70 IF M<1 OR M>12 GOTO 40
80  FOR X=1 TO M
90  READ M$
100 NEXT X
110 PRINT:PRINT M$
120 RESTORE
130 PRINT:PRINT"QUIERES SEGUIR ?"
140 INPUT A$
150 IF A$="SI" GOTO 40
160 END
```



DEF FN (de DEFine FunctioN = Definir función). Permite definir nuevas funciones matemáticas y encerrarlas en la función FN. En caso de repeticiones, por ejemplo, permite ahorrar tiempo y reducir la posibilidad de error.

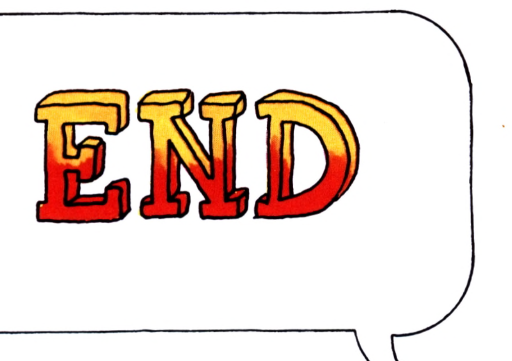
```
10 REM DEFINE UNA FUNCION COMPLICADA
20 DEF FNY (X)=(SIN(X)*X^2* (COS(2*X)+X^3)-X^5) / (X^2+1)
30 REM AHORA SE PUEDE USAR LA FUNCION CUANTAS
40 REM VECES SE QUIERA EN EL MISMO PROGRAMA
50 CLS
60 PRINT:PRINT"INTRODUCE UN VALOR PARA LA VARIABLE X"
70 INPUT X
80 Y=FNY (X)
90 Z=(FNY(X))^2
100 PRINT"X  = ";X,"Y = ";Y,"Z = ";Z
110 PRINT:PRINT"QUIERES CALCULAR OTROS VALORES ?"
120 INPUT A$
130 IF A$="SI" GOTO 60
140 END
```

DIM (de DIMension = dimensión). Indica al ordenador que dé las dimensiones a un determinado espacio de memoria y que lo conserve en cadenas de caracteres, que no se podrían contener en una «variable» individual.

¿QUEREIS QUE VUESTRO ORDENADOR ESCRIBA LOS CUADROS?

Con este programita el ordenador escribirá en la pantalla el cuadro que queráis hasta el del 10.

```
10 REM CUADROS
20 DIM T(10,10)
30 FOR J=1 TO 10
40 FOR K=1 TO 10
50 T(J,K)=J*K
60 NEXT K
70 NEXT J
80 CLS:PRINT"QUE CUADRO QUIERES VER?"
90 INPUT I
100 IF I<0 OR I>10 GOTO 80
110 PRINT
120 FOR L=1 TO 10
130 PRINT T(I,L)
140 NEXT L
150 PRINT:PRINT"QUIERES VER OTRO?"
160 INPUT A$
170 IF A$<>"SI" AND A$<>"NO" GOTO 150
180 IF A$="SI" GOTO 80
190 END
```

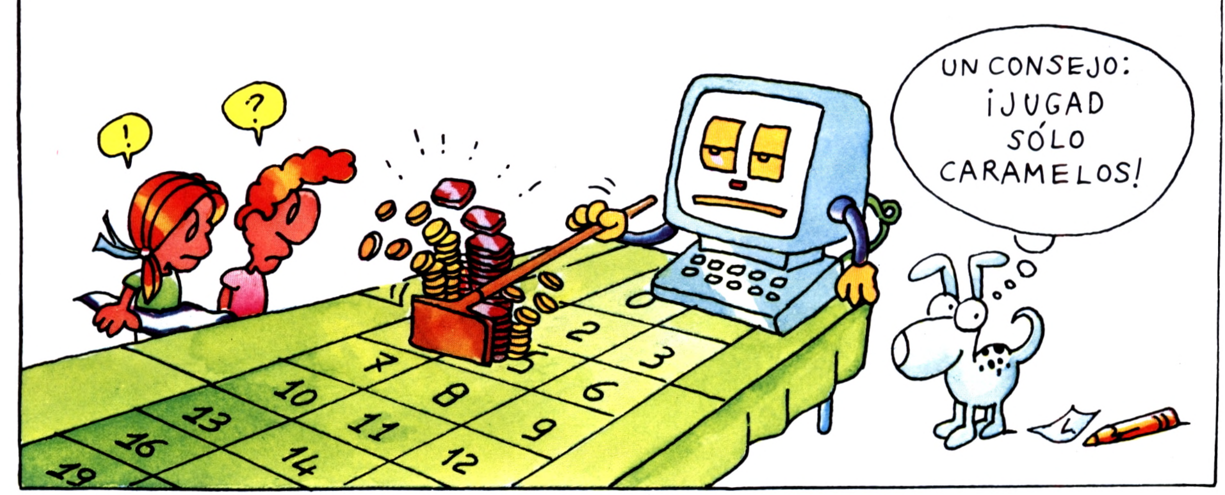


END (= fin). Indica al ordenador que el programa ha terminado y que no prosiga adelante, o sea que no haga nada más. En el lenguaje BASIC es indispensable sólo en algunos casos. Algunos ordenadores en lugar de END usan STOP.

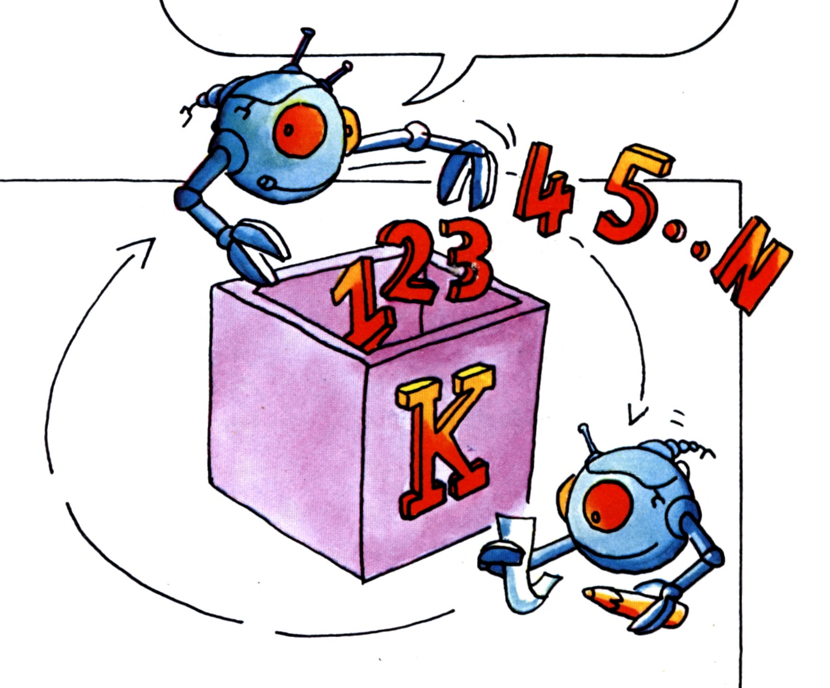
LA RULETA ELECTRONICA

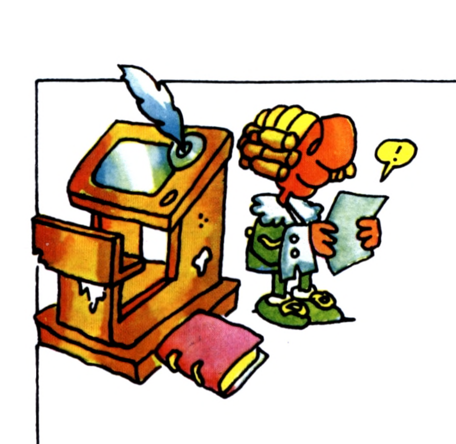
¿Tenéis suerte en el casino? Podéis verificarlo con este programita. El ordenador sacará un número entre el 0 y el 36. ¡Suerte!

```
10 REM RULETA ELECTRONICA
20 N=INT (RND*(37))
30 CLS
40 PRINT:PRINT"APUESTA A UN NUMERO ENTRE 1 Y 36"
50 INPUT T
60 IF T=N THEN PRINT"HAS GANADO !":END
70 PRINT"HAS PERDIDO !"
80 GOTO 40
90 END
```



FOR/TO/NEXT (= para/al/siguiente). El ordenador cumplirá las instrucciones puestas entre FOR, TO y NEXT hasta que la «variable de ciclo» alcance el valor indicado. En el ejemplo, la «variable de ciclo» es K y deberá alcanzar el valor N que habéis introducido.





UN PROBLEMITA PARA GAUSS

El jovencito Karl Friedrich Gauss estaba destinado a convertirse en un gran matemático. De pequeño tuvo un maestro muy antipático, que le mandó, como castigo, el ejercicio de sumar todos los números enteros comprendidos entre 1 y 100. Gauss lo resolvió con una sola operación. Vosotros se lo podéis mandar resolver al ordenador con este programita.

```
10 CLS:PRINT"SUMO LOS NUMEROS HASTA ?"
20 INPUT N
30 S=0
40 FOR K=1 TO N
50 S=S+K
60 NEXT K
70 PRINT:PRINT"LA SUMA DE LOS PRIMEROS ENTEROS VALE :";S
80 END
```

FOR TO NEXT

MIRA CÓMO SE CONSTRUYÓ EL PROGRAMA PARA VUESTRO ORDENADOR.

START

INPUT N

S=0
K+1

K>N

SÍ

NO

S=S+K

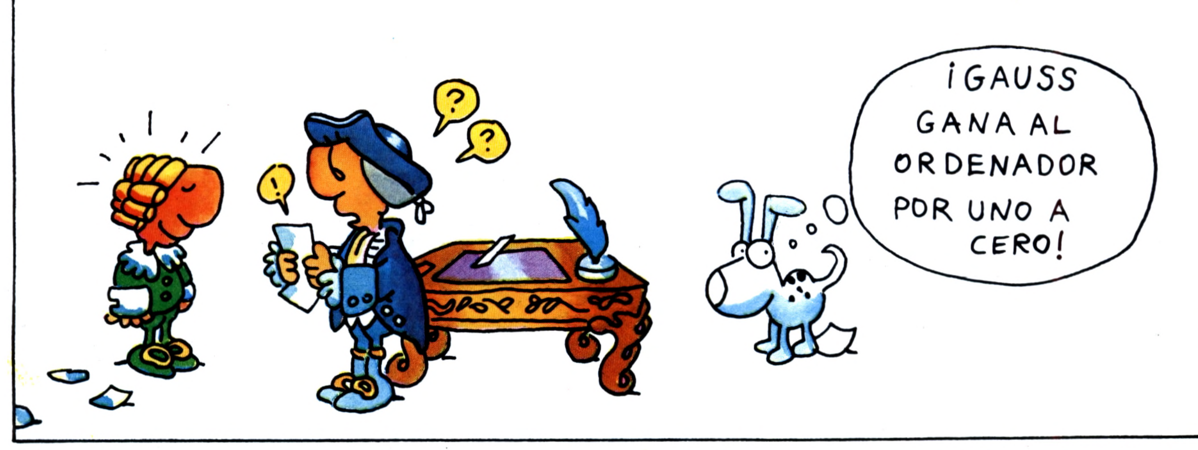
K=K+1

PRINT S

END

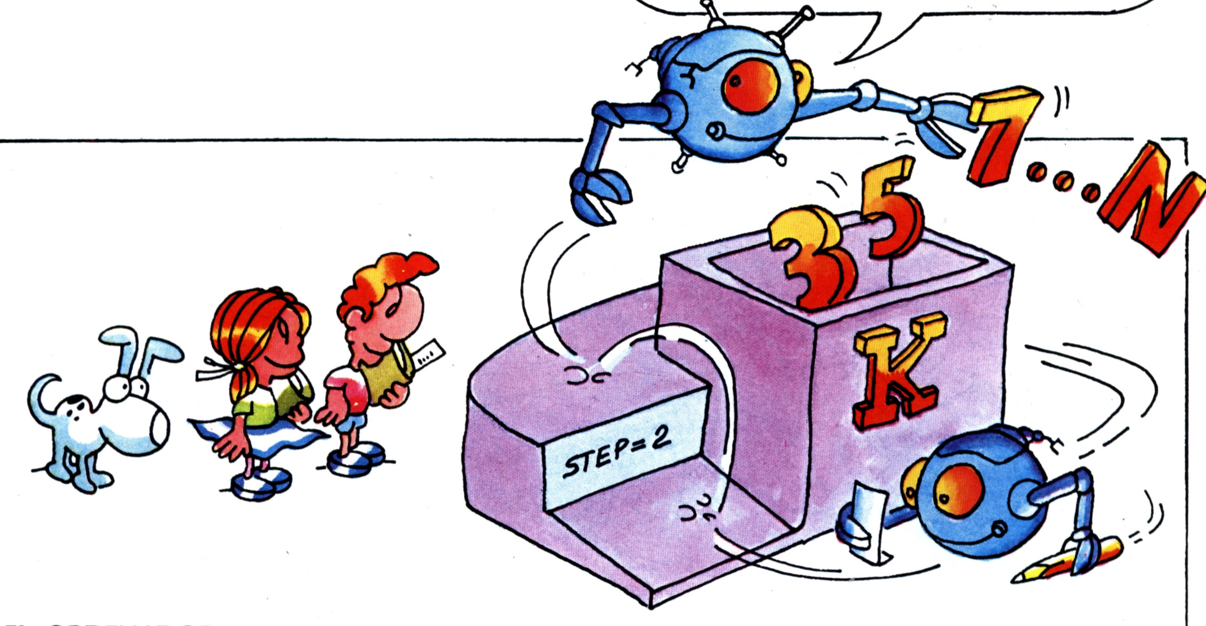
¿Y COMO RESOLVIO SU PROBLEMA EL PEQUEÑO GAUSS?

Gauss se dio cuenta de que la suma del primer número con el último, 1 y 100, del segundo con el penúltimo, 2 y 99, del tercero con el antepenúltimo, 3 y 98, y así sucesivamente, daba siempre como resultado 101. Dado que las parejas son 50, multiplicó 101×50 y resolvió el problema en un santiamén, con gran consternación del antipático maestro.



FOR/TO/STEP/NEXT (step = paso). El ordenador realizará cíclicamente las instrucciones entre FOR/TO y NEXT hasta que la «variable de ciclo» no haya alcanzado el valor indicado. El incremento es de 1, salvo que se indique el paso (STEP). En nuestro ejemplo es 2.

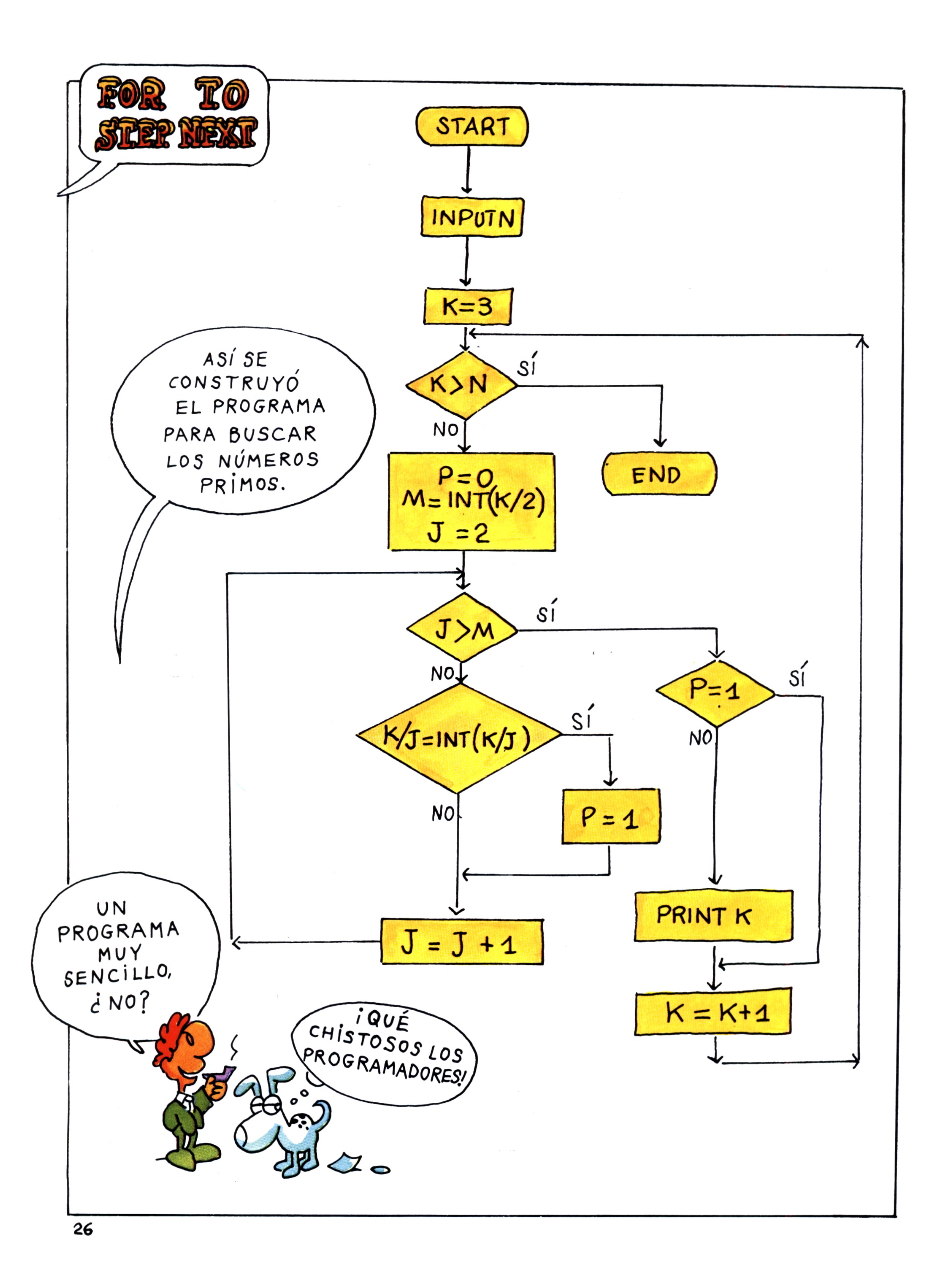




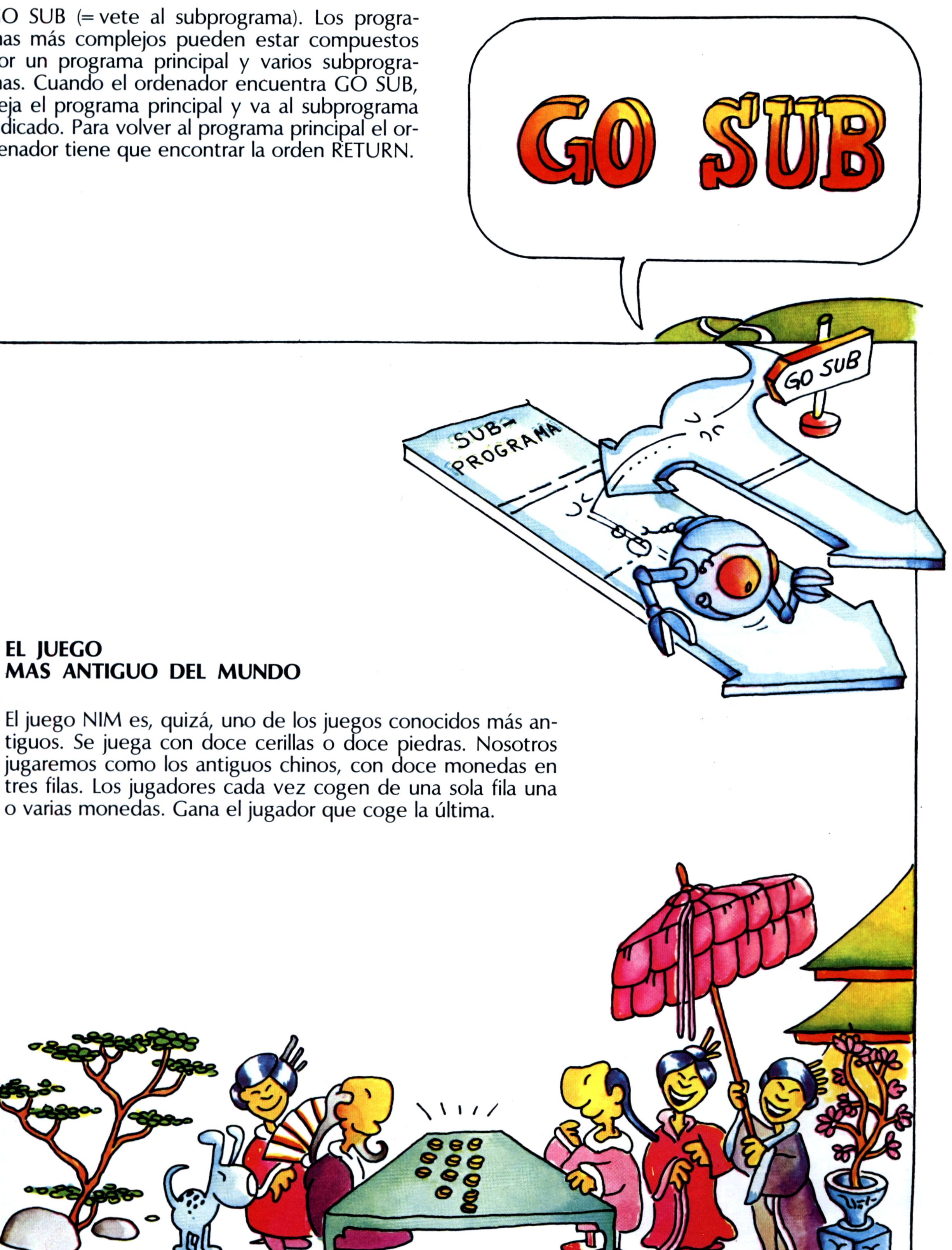
EL ORDENADOR Y LOS NUMEROS PRIMOS

Vuestro ordenador encontrará todos los números primos comprendidos entre 3 y N con este programita.

```
10 REM NUMEROS PRIMOS
20 CLS:PRINT"BUSCO LOS NUMEROS PRIMOS COMPRENDIDOS ENTRE 3 Y ?"
30 INPUT N
40 FOR K=3 TO N STEP 2
50 P=0
60 M=INT(K/2)
70 REM CONSIDERO LOS POSIBLES DIVISORES ENTRE 2 Y K/2
80 FOR J=2 TO M
90 IF K/J=INT(K/J) THEN P=1
100 NEXT J
110 IF P=1 GOTO 130
120 PRINT K
130 NEXT K
140 END
```

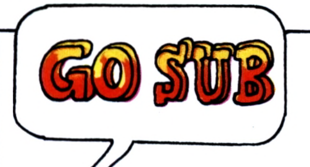

FOR TO STEP NEXT
START
INPUTN
K=3
K>N
SÍ
NO
END
ASÍ SE CONSTRUYÓ EL PROGRAMA PARA BUSCAR LOS NÚMEROS PRIMOS.
P=0
M=INT(K/2)
J =2
J>M
SÍ
NO
P=1
SÍ
NO
K/J=INT(K/J)
SÍ
NO
P = 1
PRINT K
J = J +1
K = K+1
UN PROGRAMA MUY SENCILLO, ¿NO?
¡QUÉ CHISTOSOS LOS PROGRAMADORES!

GO SUB (= vete al subprograma). Los programas más complejos pueden estar compuestos por un programa principal y varios subprogramas. Cuando el ordenador encuentra GO SUB, deja el programa principal y va al subprograma indicado. Para volver al programa principal el ordenador tiene que encontrar la orden RETURN.

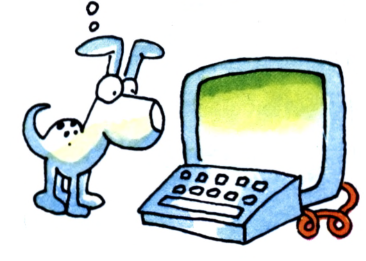


EL JUEGO MAS ANTIGUO DEL MUNDO

El juego NIM es, quizá, uno de los juegos conocidos más antiguos. Se juega con doce cerillas o doce piedras. Nosotros jugaremos como los antiguos chinos, con doce monedas en tres filas. Los jugadores cada vez cogen de una sola fila una o varias monedas. Gana el jugador que coge la última.



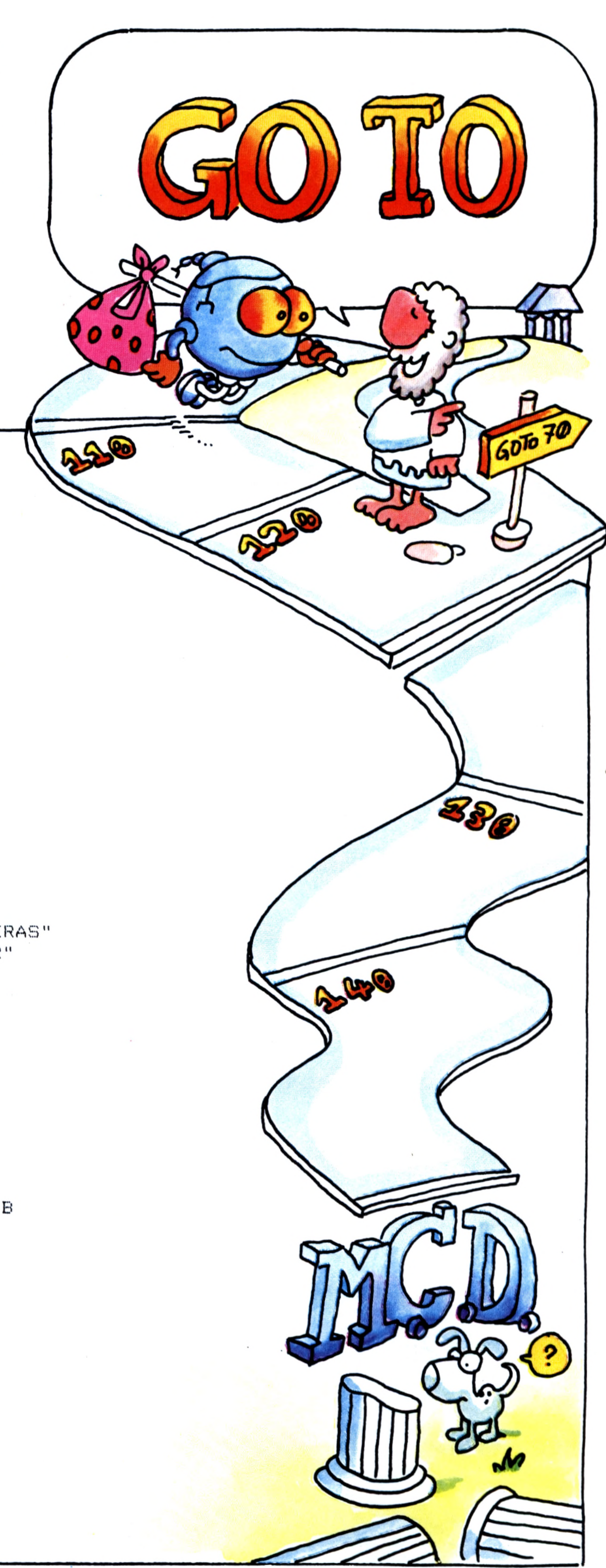

```
10 REM EL JUEGO DE NIM
20 DIM A(3)
30 A(1)=3
40 A(2)=4
50 A(3)=5
60 CLS
70 REM IMPRIME LA CONFIGURACION
80 GOSUB 380
90 PRINT:PRINT"DIME LA LINEA Y EL NUMERO DE MONEDAS"
100 INPUT F,N
110 REM CONTROLO DATOS
120 IF F>3 GOTO 170
130 IF N<=A(F) GOTO 200
140 IF A(F)=0 GOTO 170
150 PRINT:PRINT"EL NUMERO DE MONEDAS NO ES CORRECTO"
160 GOTO 90
170 PRINT:PRINT"LA LINEA NO ES VALIDA"
180 GOTO 90
190 REM COJO LAS MONEDAS
200 GOSUB 490
210 IF C=1 GOTO 240
220 PRINT:PRINT"HAS GANADO TU"
230 GOTO 340
240 GOSUB 380
250 F=INT((3*RND)+1)
260 IF A(F)=0 GOTO 250
270 REM F ES IGUAL A UNA DE LAS TRES LINEAS
280 N=INT((A(F)*RND)+1)
290 PRINT:PRINT"AHORA JUEGO YO. LINEA ";F;"MONEDAS ";N
300 GOSUB 500
310 IF C=0 GOTO 330
320 GOTO 80
330 PRINT:PRINT"HE GANADO YO"
340 PRINT"QUIERES SEGUIR JUGANDO? SI/NO"
350 INPUT R$
360 IF R$="SI" GOTO 30
370 GOTO 570
380 REM SUBPROGRAMA PARA IMPRIMIR LAS LINEAS
390 PRINT
400 FOR J=1 TO 3
410 PRINT J;"  ";
420 IF A(J)=0 GOTO 460
430 FOR K=1 TO A(J)
440 PRINT"O ";
450 NEXT K
460 PRINT
470 NEXT J
480 RETURN
490 REM SUBPROGRAMA PARA COGER LAS MONEDAS
500 C=0
510 A(F)=A(F)-N
520 FOR K=1 TO 3
530 IF A(K)=0 GOTO 550
540 C=1
550 NEXT K
560 RETURN
570 END
```



¡PERO AL ORDENADOR NO SE LE PUEDE HACER TRAMPAS!



GO TO (= vete a). Indica al ordenador que prosiga el programa a la línea indicada, que puede ser para adelante o para atrás. Se llama también instrucción de salto condicionado. Pues, cuando el ordenador la encuentra, el ordenador tiene que ir siempre a la línea indicada después de GO TO.



EUCLIDES Y EL MAXIMO COMUN DIVISOR

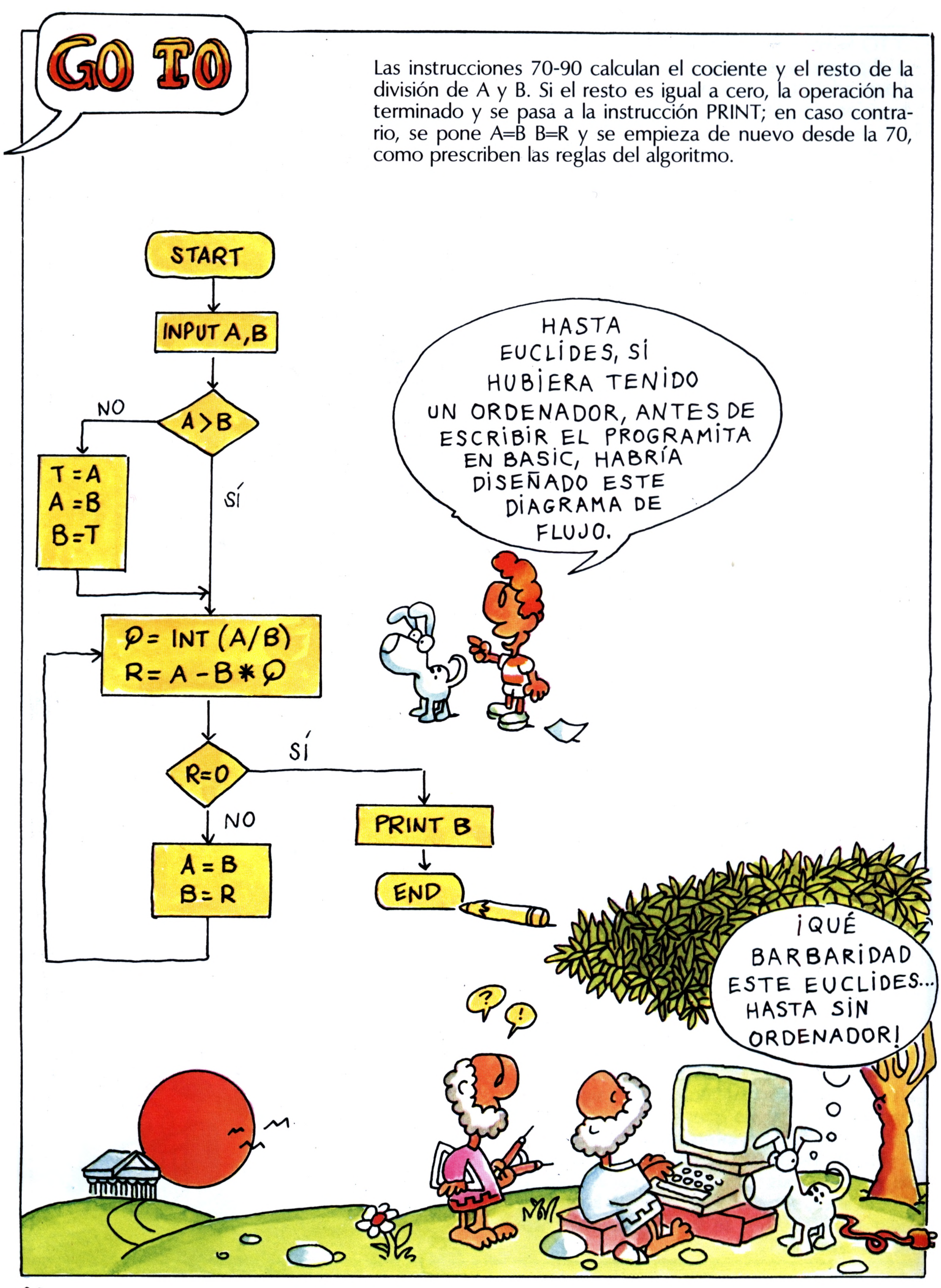
Euclides es el inventor del algoritmo para calcular el M.C.D. de dos números. Es una «receta» que se puede expresar con las siguientes reglas:

- Si A y B son dos números y A es mayor que B, se divide A por B, y R es el resto.
- Si R es igual a cero, B es el M.C.D.
- Si R no es igual a cero, se sustituye A por B, y B por R, repitiendo el proceso desde el principio.

He aquí transformados en instrucciones BASIC los pasos del algoritmo de Euclides.

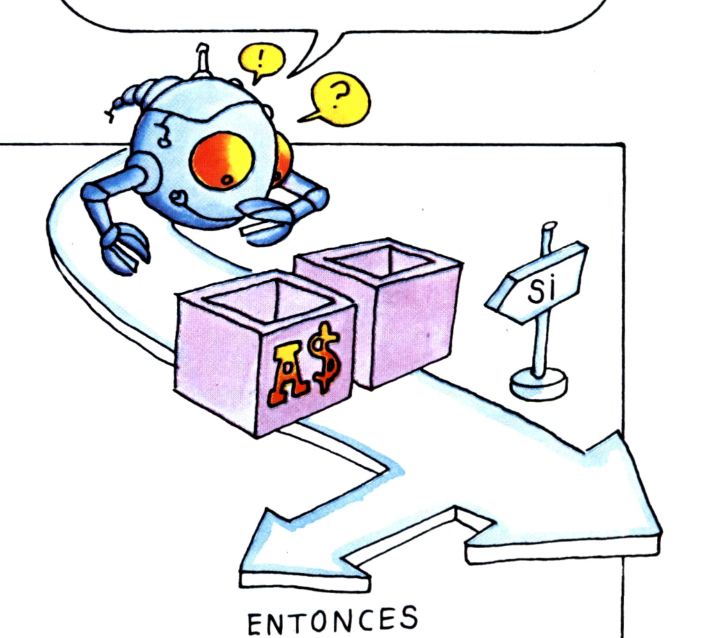
```
10 REM CALCULO DEL M.C.D.
20 CLS:PRINT"PON LOS DOS NUMEROS QUE QUIERAS"
30 PRINT"CALCULAR EL MAXIMO COMUN DIVISOR"
40 INPUT A,B
50 IF A>B GOTO 90
60 T=A
70 A=B
80 B=T
90 Q=INT(A/B)
100 R=A-B*Q
110 IF R=0 GOTO 150
120 A=B
130 B=R
140 GOTO 90
150 PRINT:PRINT"MAXIMO COMUN DIVISOR = ";B
160 END
```

La instrucción 30 pregunta si A es mayor que B; si así fuere, salta a la instrucción 70; en caso contrario continúa con las siguientes instrucciones, en las que, usando un campo de comodín T, se intercambian los valores A y B. En los BASIC más avanzados se puede escribir más de una asignación en la misma línea, separándolas con ":"; luego, en lugar de 40-50-60, se podía haber escrito: 40 T=A:A=B:N=T.


GO TO
Las instrucciones 70-90 calculan el cociente y el resto de la división de A y B. Si el resto es igual a cero, la operación ha terminado y se pasa a la instrucción PRINT; en caso contrario, se pone A=B B=R y se empieza de nuevo desde la 70, como prescriben las reglas del algoritmo.
START
INPUT A,B
A>B
NO
T=A A=B B=T
SÍ
Q= INT (A/B) R= A-B*Q
R=0
SÍ
NO
A=B B=R
PRINT B
END
HASTA EUCLIDES, SÍ HUBIERA TENIDO UN ORDENADOR, ANTES DE ESCRIBIR EL PROGRAMITA EN BASIC, HABRÍA DISEÑADO ESTE DIAGRAMA DE FLUJO.
¡QUÉ BARBARIDAD ESTE EUCLIDES... HASTA SIN ORDENADOR!

IF (= si). Pon el ordenador ante una elección que dependa de una condición. Si ésta se cumple, entonces (THEN) hace algo o va a (GO TO) otra línea. En nuestro ejemplo, cuando A$ = "si", entonces el ordenador «salta» tres líneas más abajo y escribe el animal que habíais pensado.





EL ORDENADOR ADIVINO

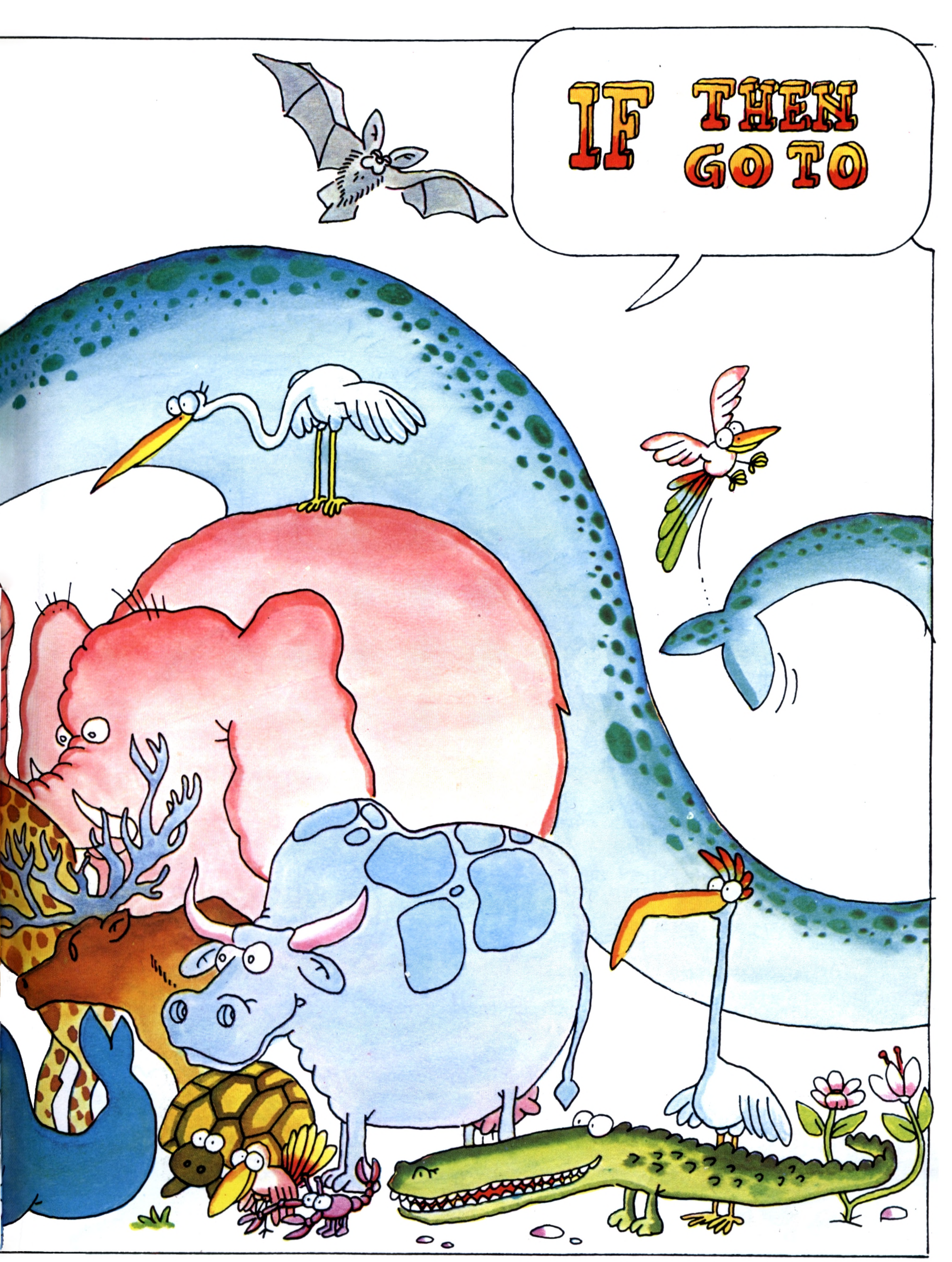
Con este programita podéis «enseñar» a vuestro ordenador que acierte un animal que habéis pensado. Ojo, fijaos bien cómo está construido. Hemos puesto también «un ciclo de espera», que no es nada más que un ciclo FOR/TO/NEXT en vacío, hecho sólo para que no espere el ordenador.

```
10 REM EL ORDENADOR ADIVINO
20 CLS:PRINT"PIENSA EN UN ANIMAL"
30 PRINT"ENTRE ESTOS TRES"
40 PRINT"ARANA  TIGRE  SERPIENTE"
50 FOR I=1 TO 3000 :NEXT I
60 PRINT"YA LO HAS PENSADO ?"
70 INPUT R$
80 IF NOT R$="SI" THEN GOTO 50
90 PRINT"CONTESTA A ESTAS PREGUNTAS"
100 PRINT"HACE TELERANAS ?"
110 INPUT A$
120 IF A$="SI" THEN GOTO 170
130 PRINT"ES UN MAMIFERO?"
140 INPUT A$
150 IF A$="SI" THEN GOTO 180
160 PRINT"HAS PENSADO EN LA SERPIENTE":GOTO 190
170 PRINT"HAS PENSADO EN LA ARANA":GOTO 190
180 PRINT"HAS PENSADO EN EL TIGRE":GOTO 190
190 END
```



HACEDLO VOSOTROS

Si habéis entendido cómo funciona el programita de la página anterior, fácilmente podréis cambiar los animales que el ordenador puede «acertar». También podréis alargar el programa, haciéndole conocer otros animales, incluso todos los dibujados en esta página. Para hacerlo aún más misterioso, os aconsejamos que coloquéis al principio del programa y antes de cada pregunta una línea con la orden CLS (o la que corresponda en vuestro ordenador). Así, cada vez se limpiará la pantalla y dará la impresión de que el ordenador dialoga con vuestros amigos. Será el primer videojuego hecho completamente por vosotros. ¡Que os divirtáis!

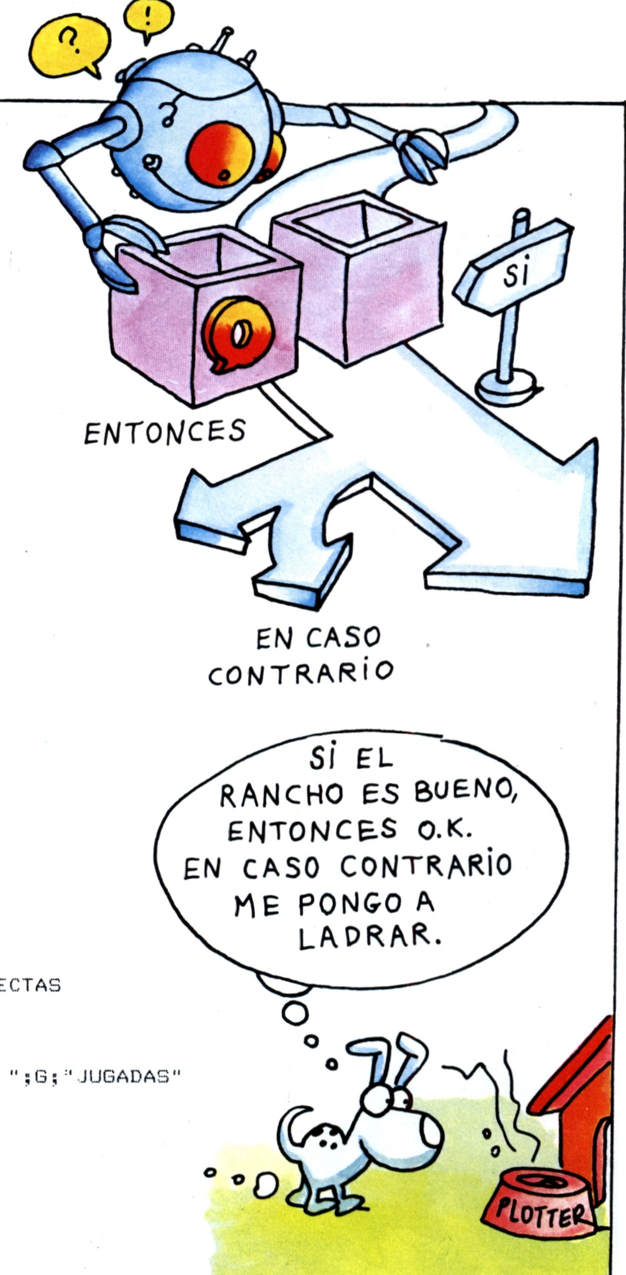

IF THEN GO TO

IF GOTO
IF THEN ELSE

ELSE (= en caso contrario). Permite al ordenador hacer dos cosas después de una elección: si (IF) se cumple la condición, entonces (THEN) hace una cosa; en caso contrario (ELSE) hace otra. En nuestro caso, si Q = 3, entonces debe ir a la línea 290; en caso contrario, a la línea 100.

OTRO JUEGO

El ordenador «piensa» un número de tres cifras y tú debes acertarlo. ¡Animo, quizá te ayude!




```
10 REM ACIERTA UN NUMERO DE TRES CIFRAS
20 DIM M(3),T(3),B$(3)
30 CLS
40 REM NUMERO DE JUGADORES ENTRE 3 Y 10
50 G=INT((RND*7)+3)
60 C=0
70 REM TRES CIFRAS AL AZAR ENTRE 0 Y 9
80 FOR K=1 TO 3
90 M(K)=INT(RND*10)
100 NEXT K
110 C=C+1
120 IF C>G THEN 320
130 PRINT:PRINT"JUGADA N.   :";C
140 PRINT"INTRODUCE TUS CIFRAS"
150 INPUT T(1),T(2),T(3)
160 REM CONTROL DE EXACTITUD
170 B$(1)="NO":B$(2)="NO":B$(3)="NO":Q=0
180 FOR P=1 TO 3
190 FOR K=1 TO 3
200 IF M(K)=T(P) AND K=P THEN 230
210 IF M(K)=T(P) AND K<>P THEN B$(P)="NO"
220 GOTO 240
230 Q=Q+1:B$(P)="SI":GOTO 250
240 NEXT K
250 NEXT P
260 REM COMPRUEBO EL RESULTADO
270 PRINT "   ";B$(1);B$(2);B$(3)
280 REM CONTROLO EL NUMERO DE CIFRAS CORRECTAS
290 IF Q=3 THEN 300 ELSE 110
300 PRINT"HAS ACERTADO EL NUMERO"
310 GOTO 340
320 PRINT"NO HAS ACERTADO EN EL LIMITE DE ";G;"JUGADAS"
330 PRINT"MI NUMERO ERA: ";M(1);M(2);M(3)
340 PRINT"QUIERES SEGUIR JUGANDO ? SI/NO"
350 INPUT S$
360 IF S$="SI" THEN 30
370 END
```

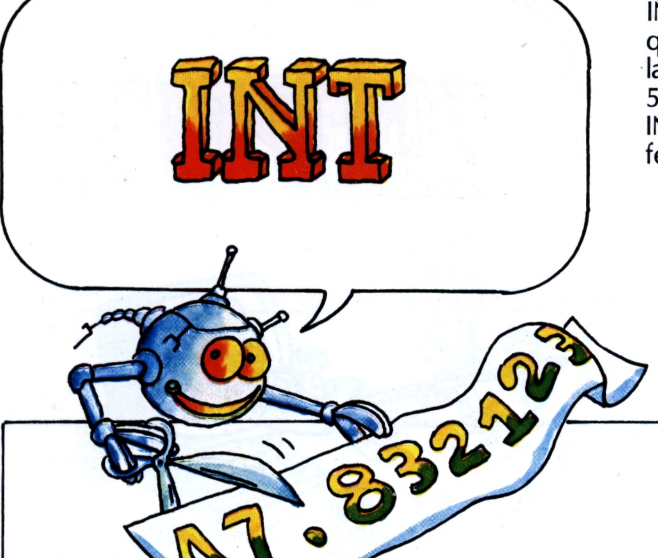
INPUT (= entrada). Indica al ordenador que pida datos de fuera. Cuando encuentra esta instrucción, se para y espera que le escribáis con el teclado los números o las palabras que necesita. En el ejemplo espera los porcentajes de anhídrido sulfuroso y la hora de las sustituciones.



PROFESIONALES DE LA CONTAMINACION

La contaminación se mide tomando a intervalos más o menos fijos la cantidad de anhídrido sulfuroso que hay en la atmósfera. El problema consiste en calcular la media de cada intervalo de tiempo y la media ponderada de las medias.
He aquí el programita que resuelve el problema:

```
10 REM CONTAMINACION
20 CLS:PRINT"CUANTAS TOMAS HAS HECHO ?"
30 INPUT N
40 REM V=VALORES TOMAS T=TIEMPOS
50 DIM V(N),T(N)
60 PRINT"INTRODUCE LOS VALORES Y LA HORA DE LAS TOMAS :"
70 FOR K=1 TO N
80 INPUT V(K),T(K)
90 NEXT K
100 S=0:PRINT
110 FOR K=1 TO N-1
120 REM MEDIA
130 M=(V(K)+V(K+1))/2
140 PRINT"MEDIA INTERVALO ";T(K);T(K+1);" = ";M
150 I=T(K+1)-T(K)
160 S=S+M*I
170 NEXT K
180 P=S/(T(N)-T(1))
190 PRINT:PRINT"MEDIA PONDERADA :";P
200 END
```



INT (de INTeger = entero). Indica al ordenador que calcule sólo la parte entera de un número, la anterior al punto. Por ejemplo, INT (57.79) es 57. Ojo, en los casos de números negativos INT calcula el valor entero inmediatamente inferior. Por ejemplo, INT (–5.3) es –6.

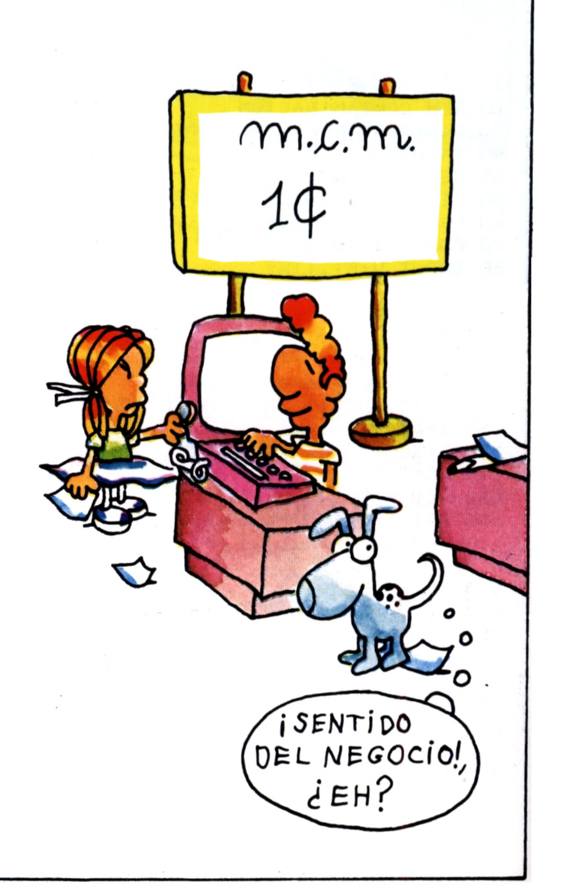
MARKO Y EL MINIMO COMUN MULTIPLO

Marko en el colegio debe calcular, a menudo, el mínimo común múltiplo de dos números. Es algo que odia. Y por esto ha construido un programa para que el ordenador lo haga por él. Ha empleado este algoritmo para construirlo:

- Si A y B son dos números en los que A es mayor que B, y A es divisible por B, entonces A es el m.c.m.
- En caso contrario, hallaremos el múltiplo más pequeño de B que sea mayor que A.
- Si éste es también múltiplo de A, habremos encontrado el m.c.m.
- En caso contrario, buscaremos el siguiente múltiplo de B repitiendo el procedimiento.

```
10 REM CALCULO DE MINIMO COMUN MULTIPLO
20 CLS:PRINT"INTRODUCE LOS DOS NUMEROS"
30 INPUT A,B
40 IF A>B GOTO 60
50 T=A:A=B:B=T
60 IF A/B=INT(A/B) GOTO 130
70 D=INT(A/B)
80 D=D+1
90 M=D*B
100 R=M-A*INT(M/A)
110 IF R<>0 GOTO 80
120 A=M
130 PRINT"MINIMO COMUN MULTIPLO:  ";A
140 END
```

La instrucción 40 intercambia los valores de A y B, en caso de que B sea mayor que A.



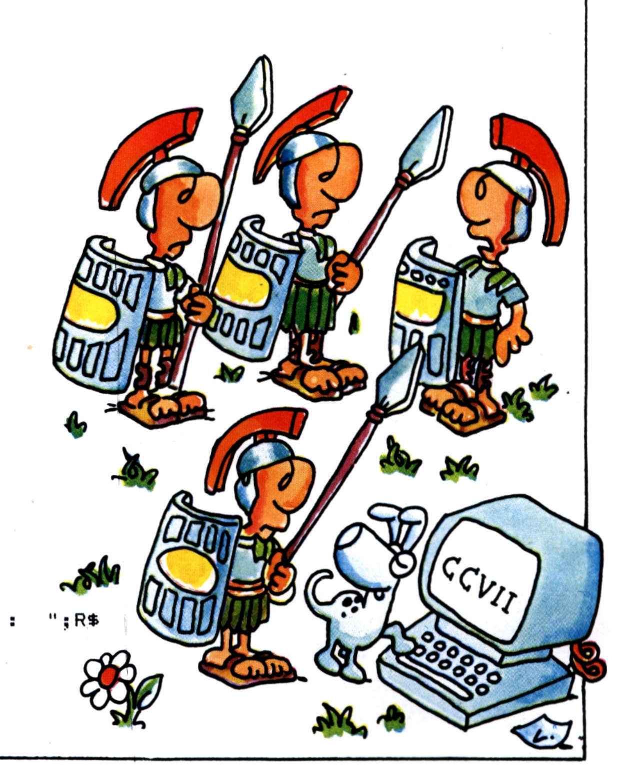
LEFT$ (= izquierda). Indica al ordenador que extraiga la parte más a la izquierda de una cadena. Si A$ = "Plotter", LEFT$ (A$,3) es "PLO". En el Spectrum LEFT$ se omite, y se escribe A$ (1 TO 3). En el ejemplo de abajo, LEFT$ extrae la primera cifra a la izquierda del número «traducido».

LOS NUMEROS ROMANOS

Este es el programa que traduce los números comprendidos entre 1 y 1.000 en las cifras que usaban los antiguos romanos.
Recordemos que:

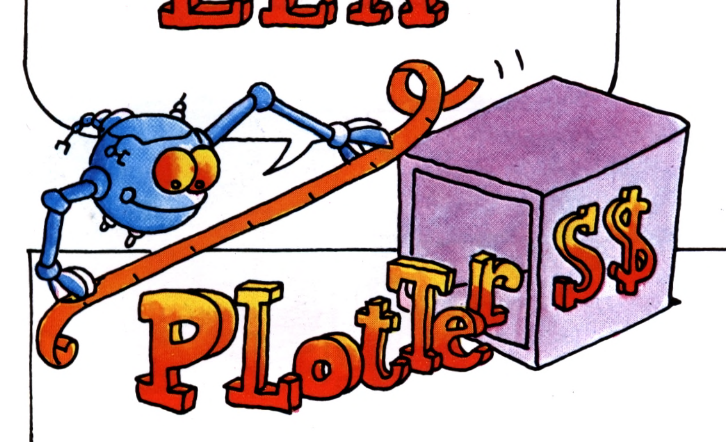
I = 1 II = 2 III = 3 IV = 4 V = 5 VI = 6 VII = 7
VIII = 8 IX = 9 X = 10 XL = 40 L = 50 LX = 60
XC = 90 C = 100 CX = 110 CL = 150
CLX = 160 D = 500 DL = 550 M = 1.000.

```
10 REM CONVERTIMOS EN NUMERO ROMANO
20 DIM V$(3,9)
30 DATA I,II,III,IV,V,VI,VII,VIII,IX
40 DATA X,XX,XXX,XL,L,LX,LXX,LXXX,XC
50 DATA C,CC,CCC,CD,D,DC,DCC,DCCC,CM
60 FOR I=1 TO 3
70 FOR J=1 TO 9
80 READ V$(I,J)
90 NEXT J
100 NEXT I
110 CLS:PRINT"INTRODUCE UN NUMERO DECIMAL"
120 PRINT"MENOR O IGUAL A 1000"
130 INPUT N
140 IF N>1000 GOTO 120
150 N$=RIGHT$(STR$(N),LEN(STR$(N))-1)
160 L=LEN(N$)
170 IF L=4 GOTO 280
180 R$=" "
190 FOR K=L TO 1 STEP -1
200 Q$=LEFT$(N$,1)
210 Q=VAL(Q$)
220 IF Q=0 GOTO 250
230 R$=R$+V$(K,Q)
240 N$=RIGHT$(N$,K-1)
250 NEXT K
260 GOTO 290
270 REM R$=M CUANDO N=1000
280 R$="M"
290 PRINT"EL NUMERO DECIMAL :  ";N
300 PRINT:PRINT"EN NUMERO ROMANO SE EXPRESA :  ";R$
310 END
```



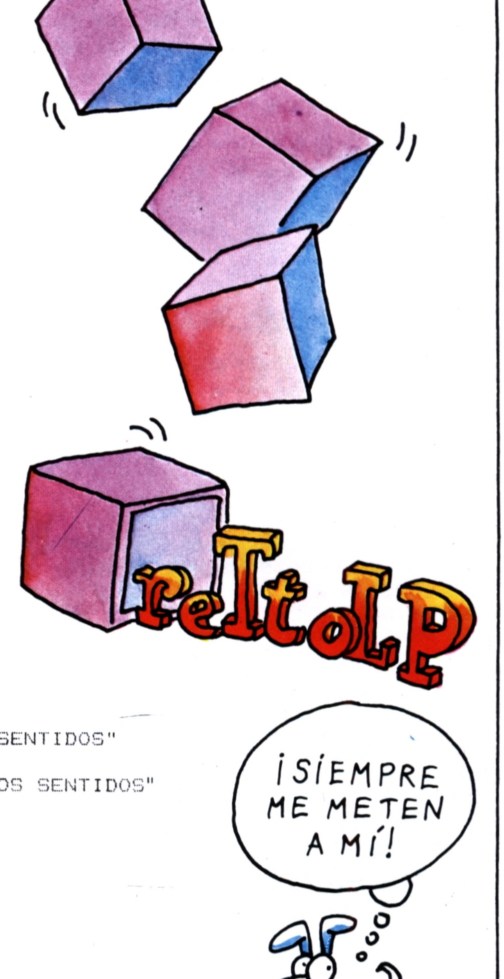
LEN

LEN (de LENght = longitud). Indica al ordenador que cuente los caracteres que hay en una cadena. Cuenta también los espacios vacíos. LEN ("W PLOTTER") es 9.
En nuestro ejemplo usamos LEN en la línea 40.



Y AHORA INVENTEMOS PALABRAS

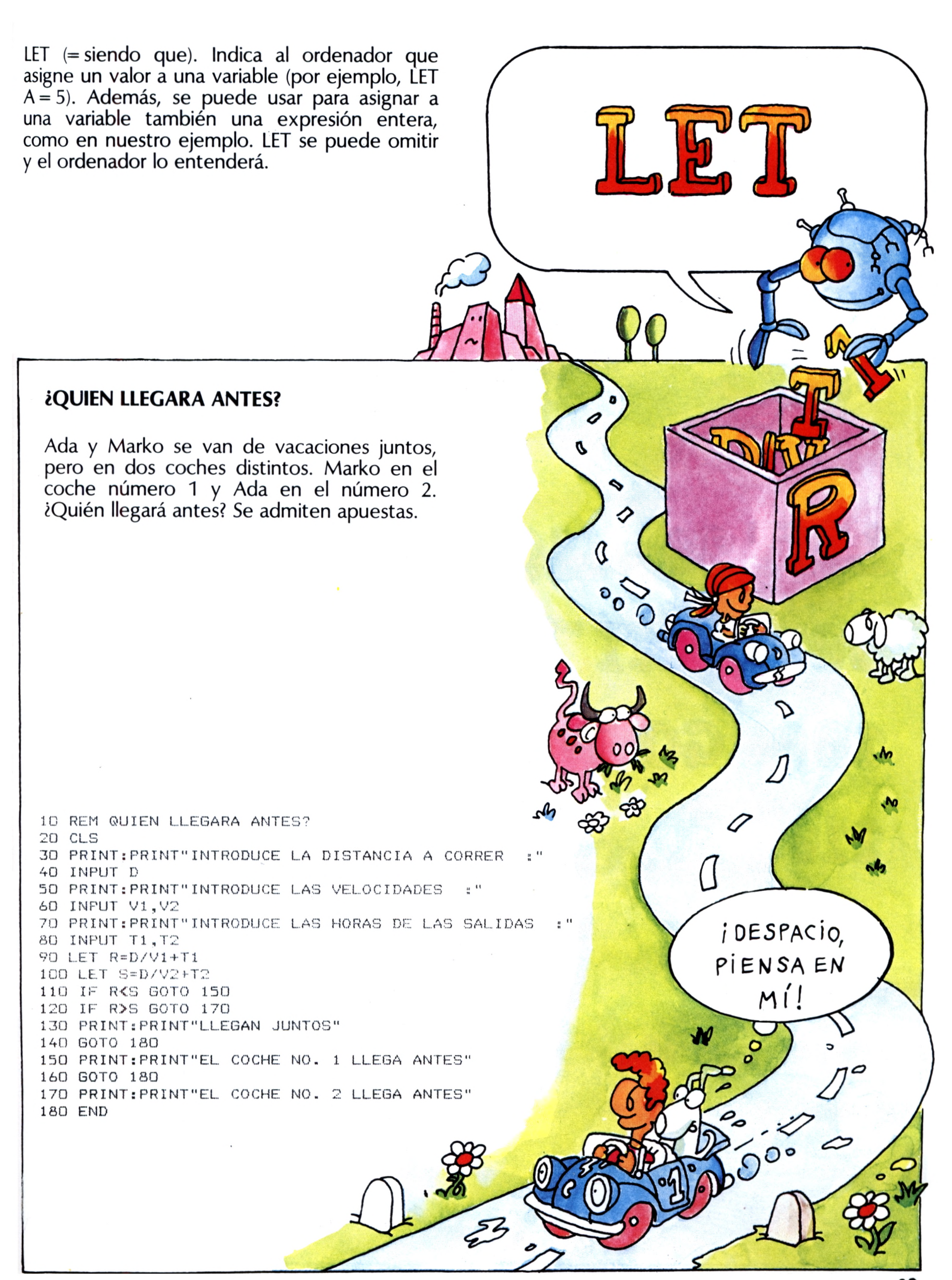
Escribamos una «cadena» de caracteres S$. Indicaremos al ordenador que la invierta y que nos diga también si es igual a sí misma leyéndola de derecha a izquierda o al revés.




```
10 CLS:PRINT"INTRODUCE UNA CADENA DE CARACTERES"
20 INPUT S$
30 A$=S$
40 L=LEN(S$)
50 FOR K=1 TO L
60 A$=LEFT$(A$,L-K+1)
70 R$=RIGHT$(A$,1)
80 B$=B$+R$
90 NEXT K
100 PRINT:PRINT"CADENA INTRODUCIDA     ";S$
110 PRINT:PRINT"CADENA INVERTIDA       ";B$
120 IF B$<>S$ GOTO 150
130 PRINT:PRINT"LA CADENA ES IDENTICA EN LOS DOS SENTIDOS"
140 GOTO 160
150 PRINT:PRINT"LA CADENA NO ES IDENTICA EN LOS DOS SENTIDOS"
160 END
```

La instrucción 40 calcula la longitud de la cadena S$. Este valor L se usa más tarde en la repetición como límite superior por K.
La instrucción 60 considera la cadena de apoyo A$ y saca (L – K + 1) los elementos más a la izquierda; la 70 pone en R$ el elemento más a la derecha de A$.

LET (= siendo que). Indica al ordenador que asigne un valor a una variable (por ejemplo, LET A = 5). Además, se puede usar para asignar a una variable también una expresión entera, como en nuestro ejemplo. LET se puede omitir y el ordenador lo entenderá.



¿QUIEN LLEGARA ANTES?

Ada y Marko se van de vacaciones juntos, pero en dos coches distintos. Marko en el coche número 1 y Ada en el número 2. ¿Quién llegará antes? Se admiten apuestas.

```
10 REM QUIEN LLEGARA ANTES?
20 CLS
30 PRINT:PRINT"INTRODUCE LA DISTANCIA A CORRER   :"
40 INPUT D
50 PRINT:PRINT"INTRODUCE LAS VELOCIDADES   :"
60 INPUT V1,V2
70 PRINT:PRINT"INTRODUCE LAS HORAS DE LAS SALIDAS   :"
80 INPUT T1,T2
90 LET R=D/V1+T1
100 LET S=D/V2+T2
110 IF R<S GOTO 150
120 IF R>S GOTO 170
130 PRINT:PRINT"LLEGAN JUNTOS"
140 GOTO 180
150 PRINT:PRINT"EL COCHE NO. 1 LLEGA ANTES"
160 GOTO 180
170 PRINT:PRINT"EL COCHE NO. 2 LLEGA ANTES"
180 END
```

LIST

LIST (= hacer una lista, «listar»). Manda al ordenador que muestre el programa. Por sí solo LIST visualiza todo el programa. Tras el número de la línea aparece la línea indicada. Inmediatamente después de las dos líneas visualiza la «lista» de líneas comprendida entre los dos números pedidos.



LIST 90 - 110

90 INT "ADA"

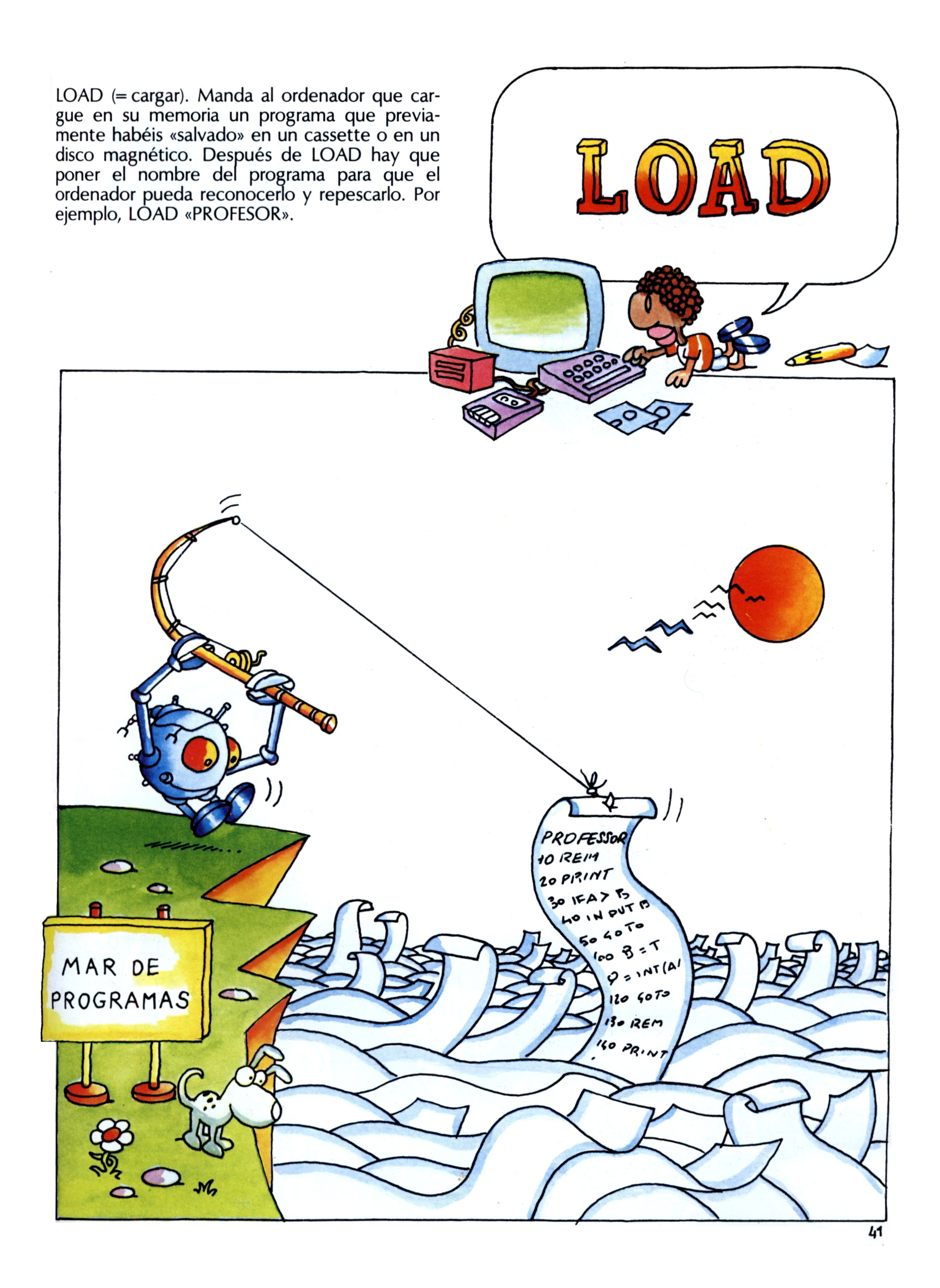
100 GO T 70

110 FO K=1



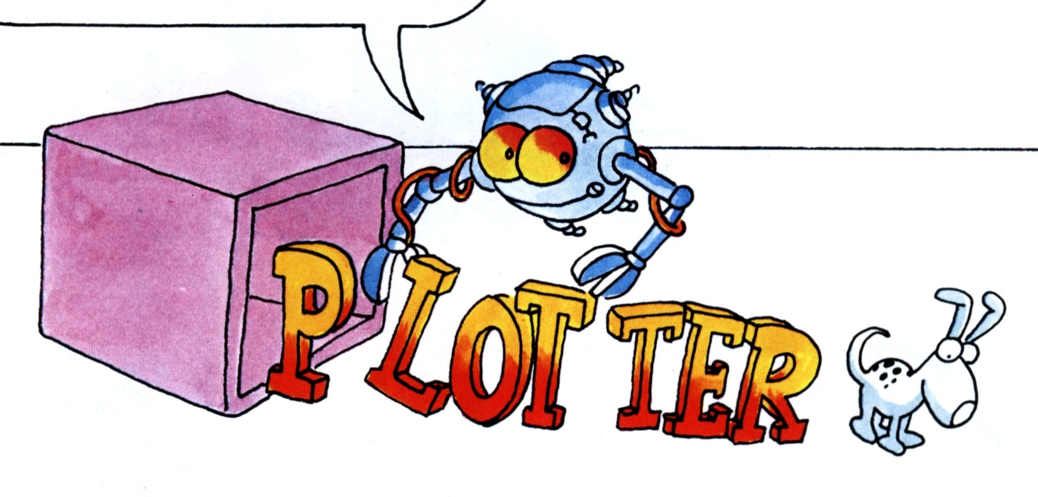


LOAD (= cargar). Manda al ordenador que cargue en su memoria un programa que previamente habéis «salvado» en un cassette o en un disco magnético. Después de LOAD hay que poner el nombre del programa para que el ordenador pueda reconocerlo y repescarlo. Por ejemplo, LOAD «PROFESOR».





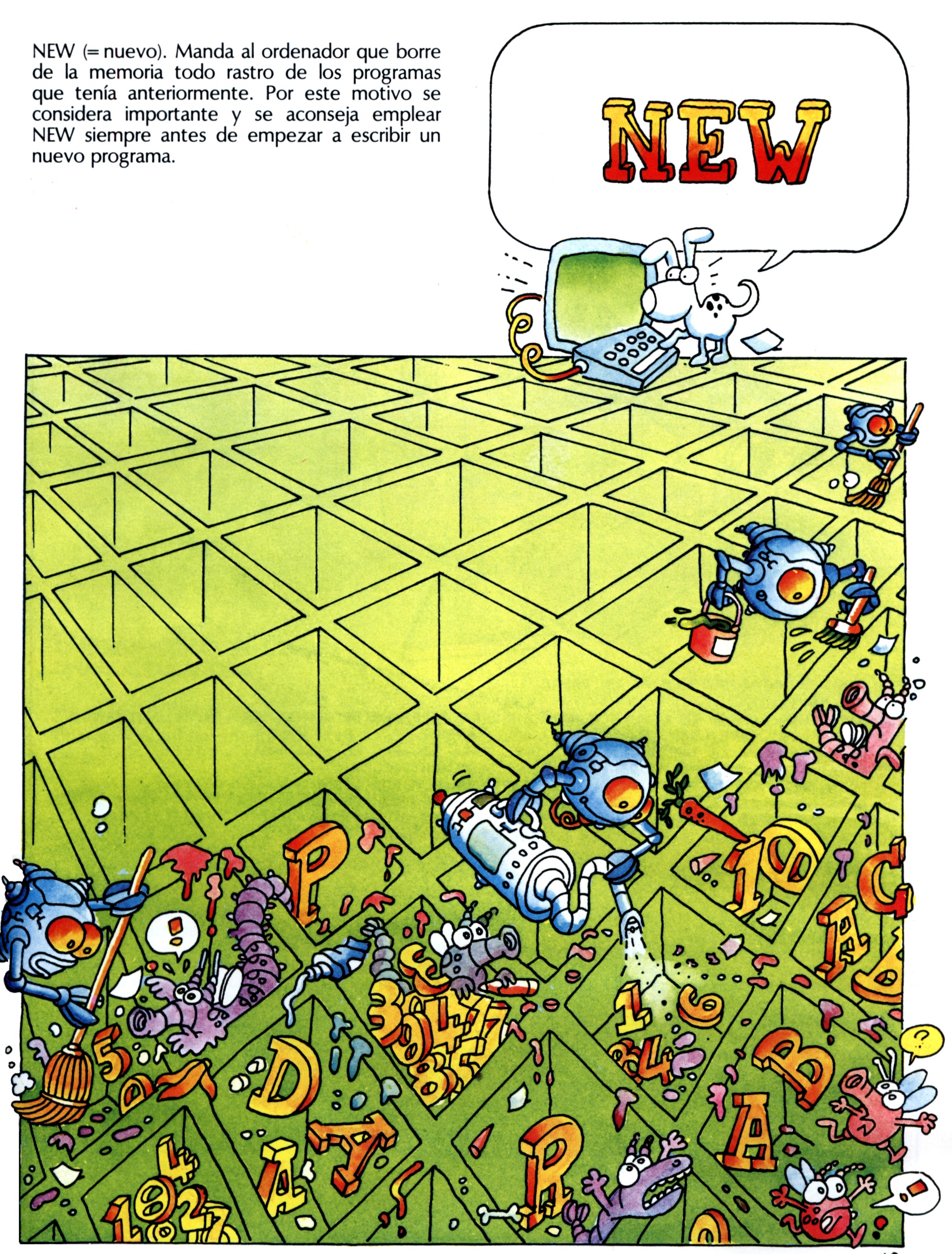
MID$ (de MIDdle = en el medio). Para sacar caracteres de una cadena. Por ejemplo, MID$ ("PLOTTER",2,3) es LOT. En el Spectrum hay que omitir MID$ y escribir sólo PLOTTER (2 TO 4). En nuestro ejercicio, MID$ saca las primeras letras de los nombres metidos en el programa y las ordena.

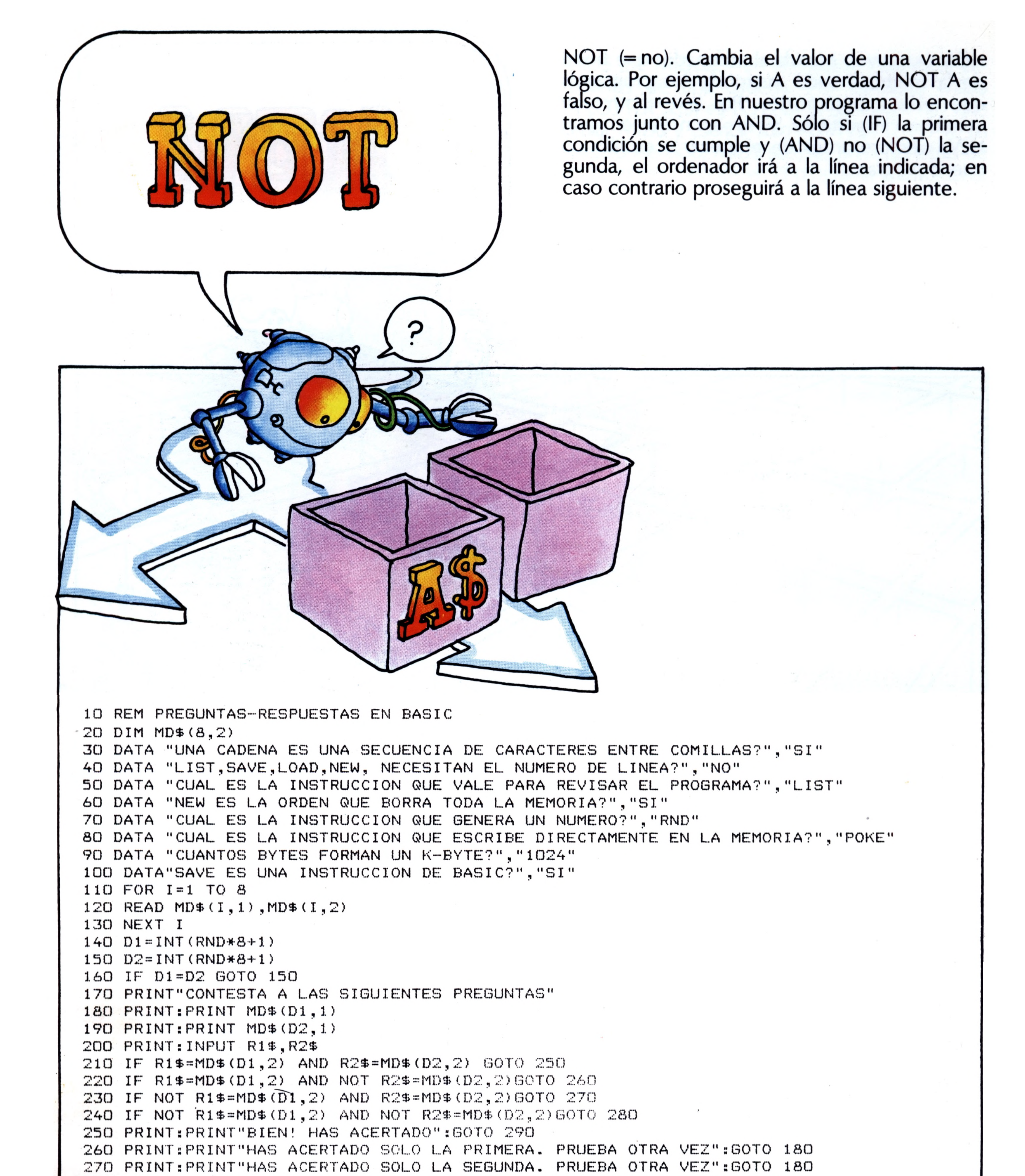



```
10 REM PROGRAMA PARA ORDENAR
20 CLS:PRINT"INTRODUCE EL NUMERO DE NOMBRES PARA ORDENAR  (MAX 30)"
30 INPUT NM
40 IF NM<1 OR NM >30 GOTO 20
50 DIM MN$(NM)
60 FOR I=1 TO NM
70 PRINT"INTRODUCE EL NOMBRE NUMERO";I;"(MAX 24 CARACTERES)"
80 INPUT MN$(I)
90 IF LEN(MN$(I))>24 GOTO 70
100 NEXT I
110 FL=0
120 FOR I=1 TO NM - 1
130 FOR J=1 TO 24
140 A$=MID$(MN$(I),J,1):B$=MID$(MN$(I+1),J,1)
150 IF A$="" OR B$="" GOTO 205
160 A=ASC(A$):B=ASC(B$)
170 IF B>A GOTO 205
180 IF A=B THEN GOTO 205
190 IF B<A THEN GOSUB 280
195 GOTO 210
200 NEXT J
205 NEXT I
210 IF FL=1 GOTO 110
220 CLS
230 PRINT" *--- NOMBRES ORDENADOS ALFABETICAMENTE ---*":PRINT
240 FOR I=1 TO NM
250 PRINT"        ";MN$(I)
260 NEXT I
270 END
280 S$=MN$(I)
290 MN$(I)=MN$(I+1)
300 MN$(I+1)=S$
310 FL=1
320 RETURN
```

ADA
ANA
ANDRÉS
AUGUSTO
BEATRIZ...

NEW (= nuevo). Manda al ordenador que borre de la memoria todo rastro de los programas que tenía anteriormente. Por este motivo se considera importante y se aconseja emplear NEW siempre antes de empezar a escribir un nuevo programa.



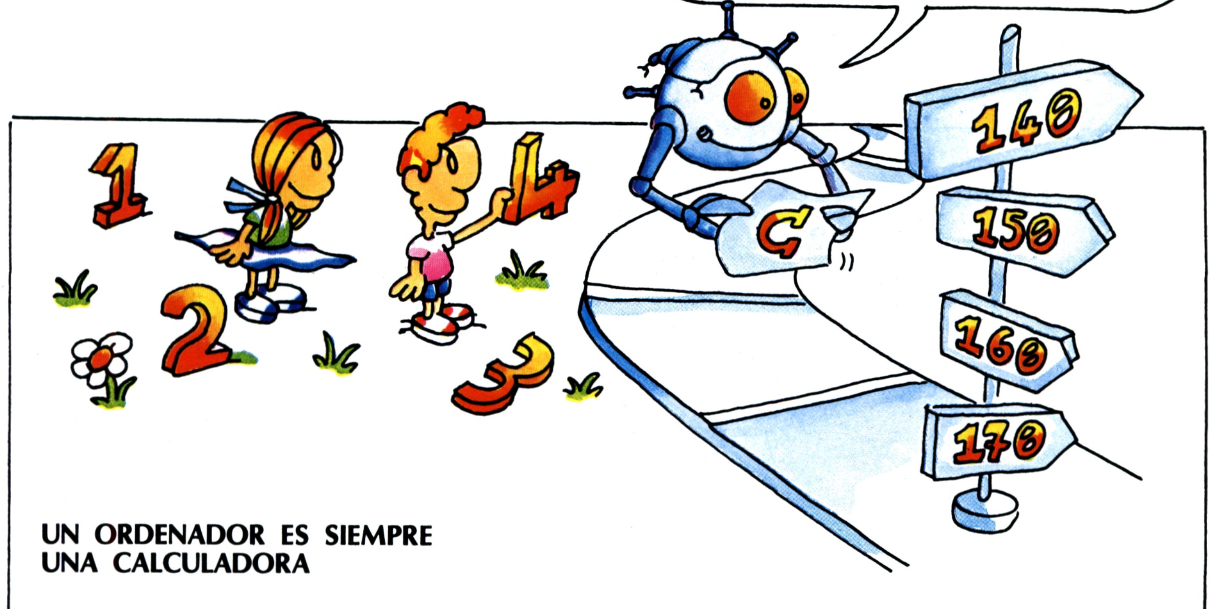


NOT (= no). Cambia el valor de una variable lógica. Por ejemplo, si A es verdad, NOT A es falso, y al revés. En nuestro programa lo encontramos junto con AND. Sólo si (IF) la primera condición se cumple y (AND) no (NOT) la segunda, el ordenador irá a la línea indicada; en caso contrario proseguirá a la línea siguiente.

```
10 REM PREGUNTAS-RESPUESTAS EN BASIC
20 DIM MD$(8,2)
30 DATA "UNA CADENA ES UNA SECUENCIA DE CARACTERES ENTRE COMILLAS?","SI"
40 DATA "LIST,SAVE,LOAD,NEW, NECESITAN EL NUMERO DE LINEA?","NO"
50 DATA "CUAL ES LA INSTRUCCION QUE VALE PARA REVISAR EL PROGRAMA?","LIST"
60 DATA "NEW ES LA ORDEN QUE BORRA TODA LA MEMORIA?","SI"
70 DATA "CUAL ES LA INSTRUCCION QUE GENERA UN NUMERO?","RND"
80 DATA "CUAL ES LA INSTRUCCION QUE ESCRIBE DIRECTAMENTE EN LA MEMORIA?","POKE"
90 DATA "CUANTOS BYTES FORMAN UN K-BYTE?","1024"
100 DATA"SAVE ES UNA INSTRUCCION DE BASIC?","SI"
110 FOR I=1 TO 8
120 READ MD$(I,1),MD$(I,2)
130 NEXT I
140 D1=INT(RND*8+1)
150 D2=INT(RND*8+1)
160 IF D1=D2 GOTO 150
170 PRINT"CONTESTA A LAS SIGUIENTES PREGUNTAS"
180 PRINT:PRINT MD$(D1,1)
190 PRINT:PRINT MD$(D2,1)
200 PRINT:INPUT R1$,R2$
210 IF R1$=MD$(D1,2) AND R2$=MD$(D2,2) GOTO 250
220 IF R1$=MD$(D1,2) AND NOT R2$=MD$(D2,2)GOTO 260
230 IF NOT R1$=MD$(D1,2) AND R2$=MD$(D2,2)GOTO 270
240 IF NOT R1$=MD$(D1,2) AND NOT R2$=MD$(D2,2)GOTO 280
250 PRINT:PRINT"BIEN! HAS ACERTADO":GOTO 290
260 PRINT:PRINT"HAS ACERTADO SOLO LA PRIMERA. PRUEBA OTRA VEZ":GOTO 180
270 PRINT:PRINT"HAS ACERTADO SOLO LA SEGUNDA. PRUEBA OTRA VEZ":GOTO 180
280 PRINT:PRINT"LAS DOS ESTAN EQUIVOCADAS. PRUEBA OTRA VEZ":GOTO 180
290 PRINT:PRINT"QUIERES CONTESTAR OTRAS PREGUNTAS?"
300 INPUT A$
310 IF A$="SI"GOTO 140
320 END
```

ON GO TO (= en caso de que, vete a). Indica al ordenador que «salte» a una de las líneas indicadas en función de un parámetro. En nuestro ejemplo, el parámetro es C. Si C = 1, el ordenador va a la línea 140; si C = 2, va a la línea 150; si C = 3, va a la línea 160; si C = 4, a la línea 170.

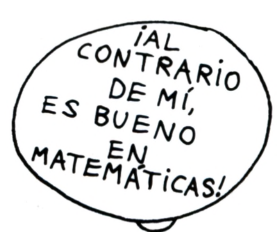




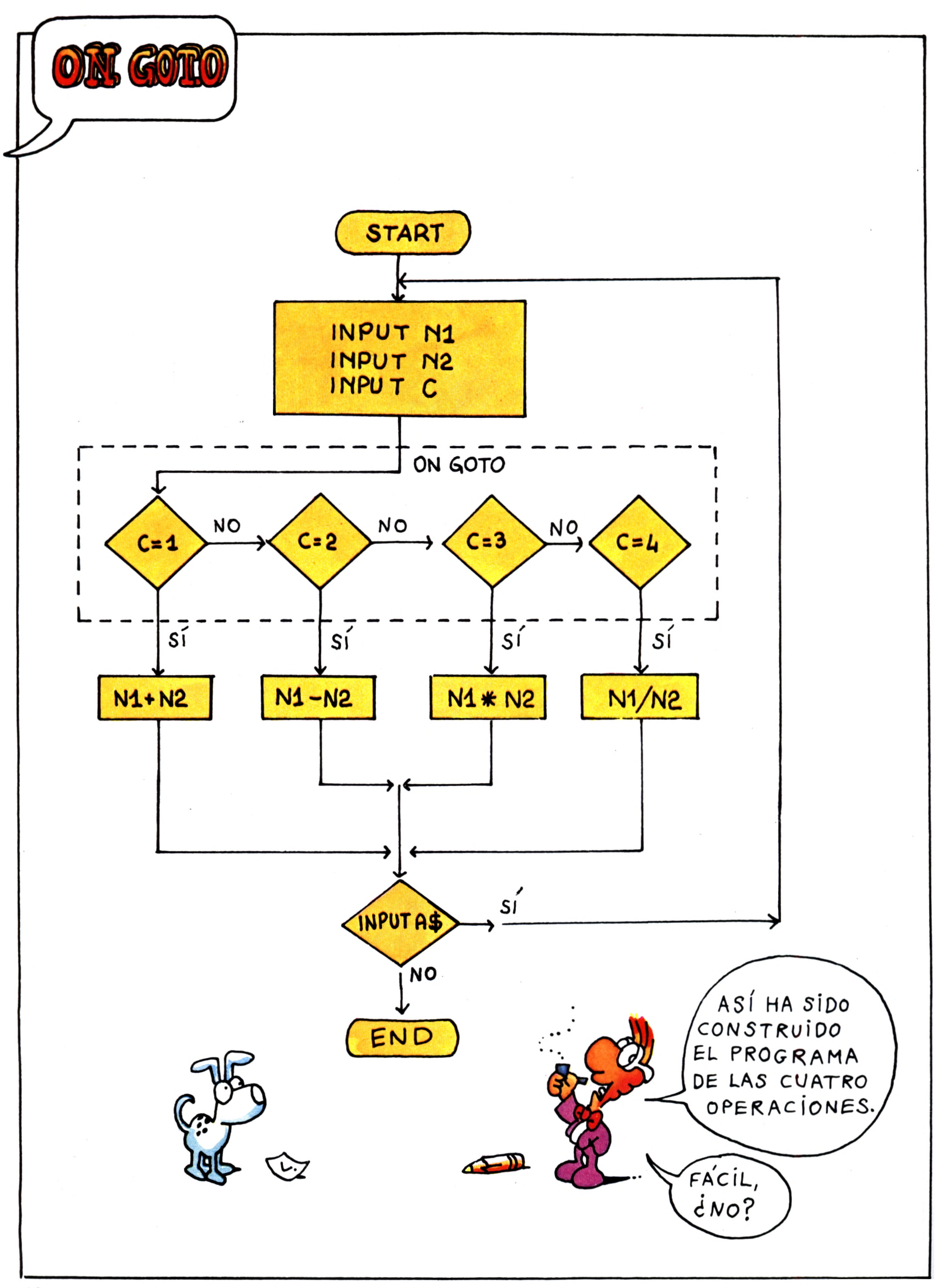
UN ORDENADOR ES SIEMPRE UNA CALCULADORA

"Volved para atrás" este programa. Meted dos números y, en función de lo que queráis, el ordenador calculará la suma, la resta, la multiplicación o la división.

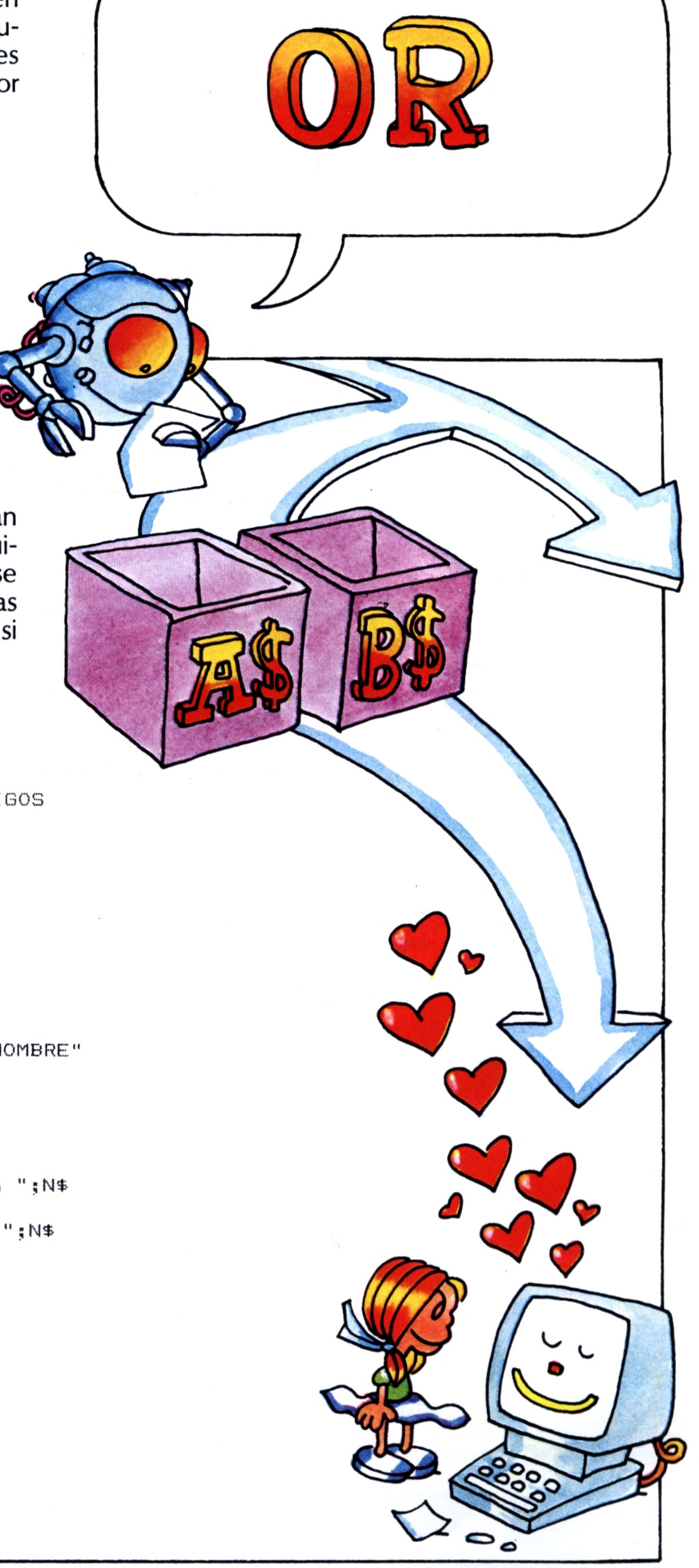
```
10 REM LA CALCULADORA
20 CLS:PRINT"INTRODUCE EL PRIMER NUMERO"
30 INPUT N1
40 PRINT:PRINT"INTRODUCE EL SEGUNDO NUMERO"
50 INPUT N2
60 CLS:PRINT"QUE OPERACION QUIERES?":PRINT
70 PRINT"        PULSA   1       PARA SUMAR":PRINT
80 PRINT"        PULSA   2       PARA RESTAR":PRINT
90 PRINT"        PULSA   3       PARA MULTIPLICAR":PRINT
100 PRINT"       PULSA   4       PARA DIVIDIR":PRINT
110 INPUT C
120 PRINT
130 ON C GOTO 150,160,170,180
140 REM REALIZO LOS CALCULOS
150 PRINT N1;" + ";N2;" = ";N1+N2:GOTO 190
160 PRINT N1;" - ";N2;" = ";N1-N2:GOTO 190
170 PRINT N1;" * ";N2;" = ";N1*N2:GOTO 190
180 PRINT N1;" : ";N2;" = ";N1/N2:GOTO 190
190 PRINT:PRINT"QUIERES HACER OTRA OPERACION?"
200 INPUT A$
210 IF A$="SI" GOTO 10
220 END
```






ON GOTO
START
INPUT N1
INPUT N2
INPUT C
ON GOTO
C=1
NO
C=2
NO
C=3
NO
C=4
SÍ
SÍ
SÍ
SÍ
N1+N2
N1-N2
N1*N2
N1/N2
INPUT A$
SÍ
NO
END
ASÍ HA SIDO CONSTRUIDO EL PROGRAMA DE LAS CUATRO OPERACIONES.
FÁCIL, ¿NO?

OR (= o). Es una «conjunción» también en BASIC. Si (IF) se realiza una condición u (OR) otra, entonces (THEN) el ordenador hace algo; en caso contrario pasa a la línea siguiente. En resumen, basta que sólo una de las condiciones puestas sea verdadera para que el ordenador cumpla la instrucción contenida en la línea.



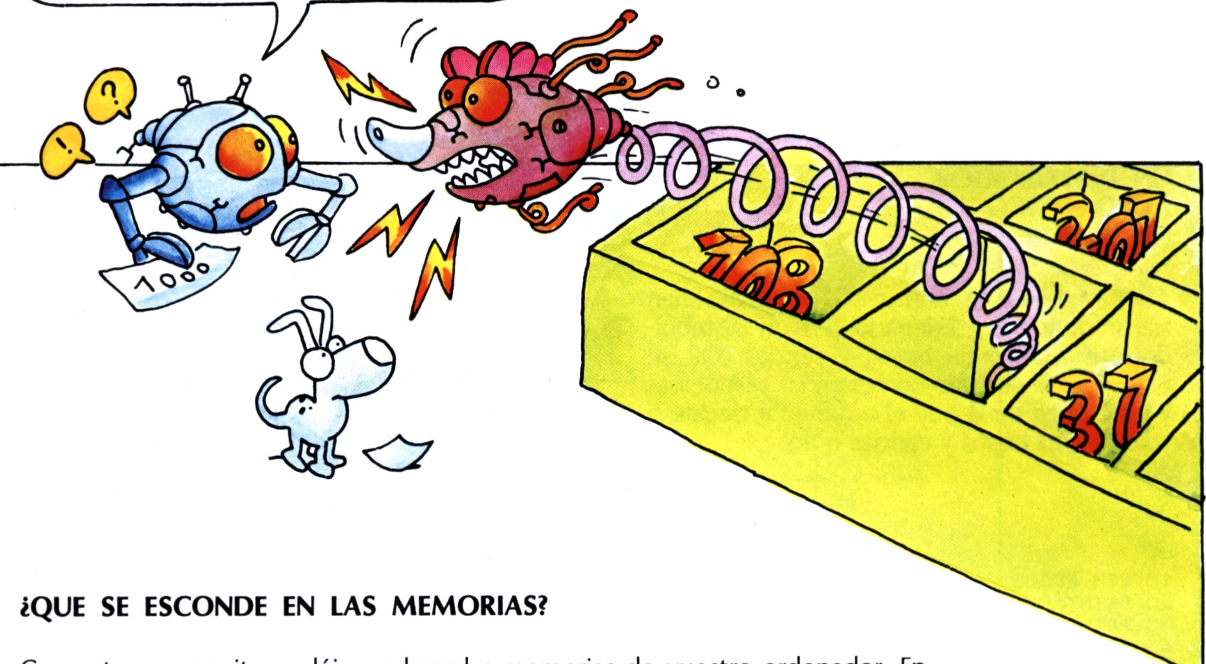
UN DISTRIBUIDOR AUTOMATICO DE MIMOS

ADA y MARKO con este programa han transformado su ordenador en un distribuidor automático de besos y caricias. OR se usa en las líneas que controlan las respuestas y en la línea en la que el ordenador decide si «distribuye» besos y caricias o no.

```
10 REM BESOS Y CARICIAS SOLO A LOS AMIGOS
20 PRINT"PLOTTER.TE GUSTA?"
30 PRINT"RESPONDE SI O NO"
40 INPUT A$
50 IF A$="SI" OR A$="NO" GOTO 70
60 GOTO 30
70 PRINT"PREFIERES A ADA Y A MARKO?"
80 PRINT"RESPONDE SI O NO"
90 INPUT B$
100 IF B$="SI" OR B$="NO" GOTO 120
110 GOTO 80
120 PRINT"COMO TE LLAMAS? ESCRIBE TU NOMBRE"
130 INPUT N$
140 IF A$="SI" OR B$="SI" GOTO 160
150 PRINT"NO ME GUSTAS ";N$:END
160 PRINT"1 BESO Y 1 CARICIA A ";N$
170 FOR K=2 TO 20
180 PRINT K;" BESOS Y ";K;" CARICIAS A ";N$
190 NEXT K
200 PRINT"ESPERO QUE SEAN SUFICIENTES ";N$
210 END
```



PEEK (to peek = ver a través de una ranura). Permite leer directamente el contenido de una celdilla de memoria. Si, por ejemplo, con PRINT PEEK (1000) conseguimos 123, significa que en la celdilla de memoria número 1000 se encuentra el valor 123, que puede representar un dato o una instrucción.

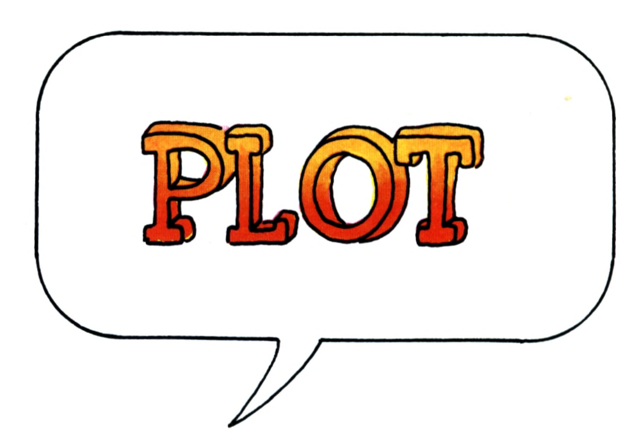


¿QUE SE ESCONDE EN LAS MEMORIAS?

Con este programita podéis explorar las memorias de vuestro ordenador. Encontraréis celdillas vacías (valor 0) o valores comprendidos entre 1 y 255. Estos valores muestran letras, símbolos gráficos, números o instrucciones en «lenguaje máquina».

```
10 REM LECTURA DE LA MEMORIA
20 CLS:PRINT"ESTE PROGRAMA PERMITE LEER"
30 PRINT"LA MEMORIA DE LA CALCULADORA":PRINT
40 PRINT"DE CUANTOS K-BYTES ES LA MEMORIA DE TU CALCULADORA?"
50 INPUT KM
60 M=KM*1024-1
70 CLS:PRINT:PRINT"INTRODUCE LA DIRECCION INICIAL ENTRE 0 Y ";M
80 INPUT S
90 IF S<0 OR S>M GOTO 70
100 PRINT:PRINT"INTRODUCE LA DIRECCION FINAL ENTRE";S;" Y ";M
110 INPUT F
120 IF F<S OR F>M GOTO 100
130 IF F<S GOTO 100
140 FOR I=S TO F
150 PRINT"DIRECCION = ";I;"    VALOR IN MEMORIA = ";PEEK(I)
160 NEXT I
170 PRINT:PRINT"DESEAS VER OTRA MEMORIA?"
180 INPUT A$
190 IF A$="SI" GOTO 70
200 END
```

PLOT (= dibujar un trazo). Enciende en la pantalla un punto luminoso de coordenadas X, Y. El color del punto está predeterminado por la instrucción COLOR seguida del número de código color. PLOT (y también COLOR) no es común a todos los ordenadores. Se utiliza para realizar gráficos, dibujos y obras de ordenador art.

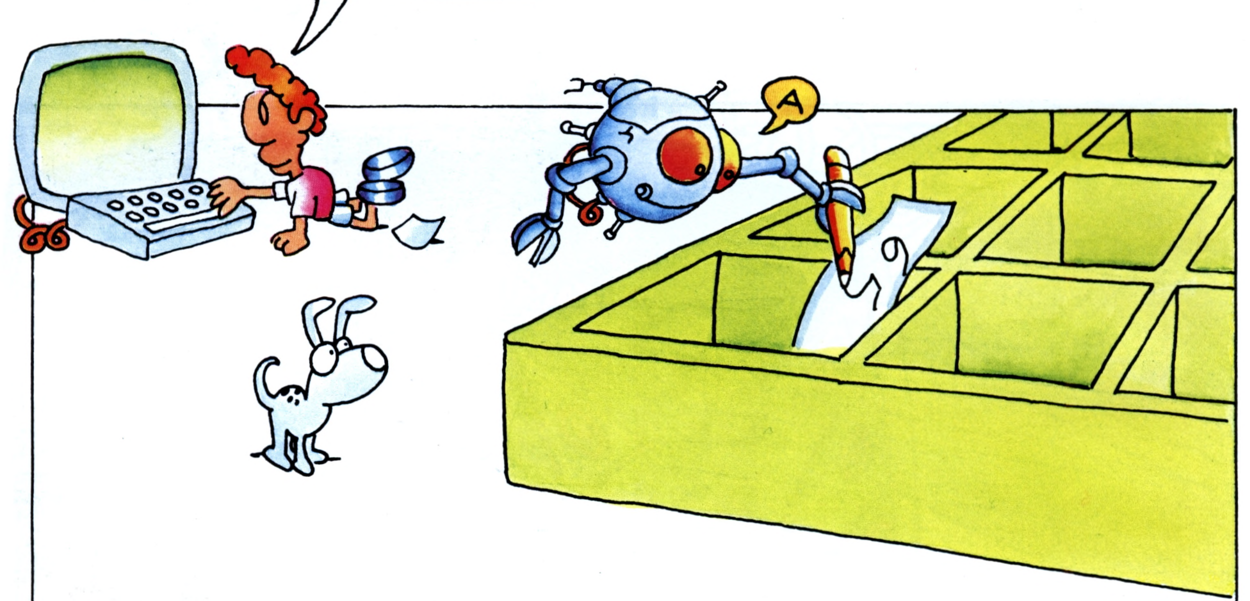


60 COLOR 4

70 PLOT 5,3



POKE (to poke = meter, introducir). Permite escribir directamente en una celdilla de memoria un valor (comprendido entre 0 y 255), que puede representar, en el código ASCII, un dato o una instrucción. Por ejemplo, POKE 15000,65 coloca en la celdilla 15000 la letra A (65 es el código ASCII de A).

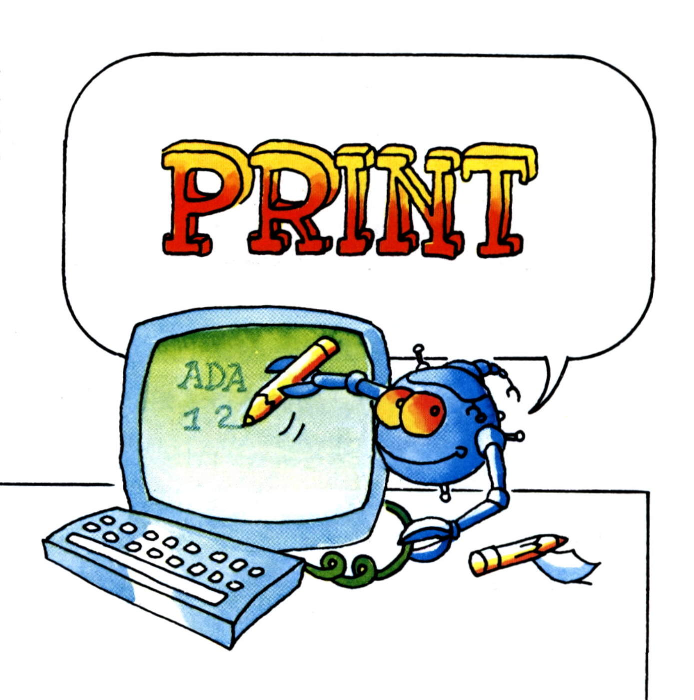


«ESCRIBIR» EN LAS MEMORIAS

Conociendo las direcciones (que dependen del plano de las memorias) y los valores exactos (siempre comprendidos entre 0 y 255), podréis hacer realizar al ordenador las cosas más variadas. Con POKE, por ejemplo, podréis crear música o dibujos en la pantalla. Yendo a ciegas os puede suceder de todo, hasta que el ordenador se pare. No tengáis miedo. En caso de problemas, apagadlo y empezad de nuevo o pasad a otro programa.

```
10 REM ESCRITURA EN LA MEMORIA
20 CLS:PRINT:PRINT
30 PRINT"CUANTAS K-BYTES DE MEMORIA TIENE TU ORDENADOR ?"
40 INPUT KB
50 B=KB*1024-1
60 PRINT"INTRODUCE EL LUGAR EN QUE QUIERES ESCRIBIR ENTRE 0 Y ";B
70 INPUT L
80 IF L<0 OR L>B GOTO 60
90 PRINT"INTRODUCE EL VALOR QUE QUIERES PONER EN LA DIRECCION ";L
100 PRINT"COMPRENDIDO ENTRE 0 Y 255"
110 INPUT V
120 IF V<0 OR V>255 GOTO 90
130 POKE L,V
140 END
```

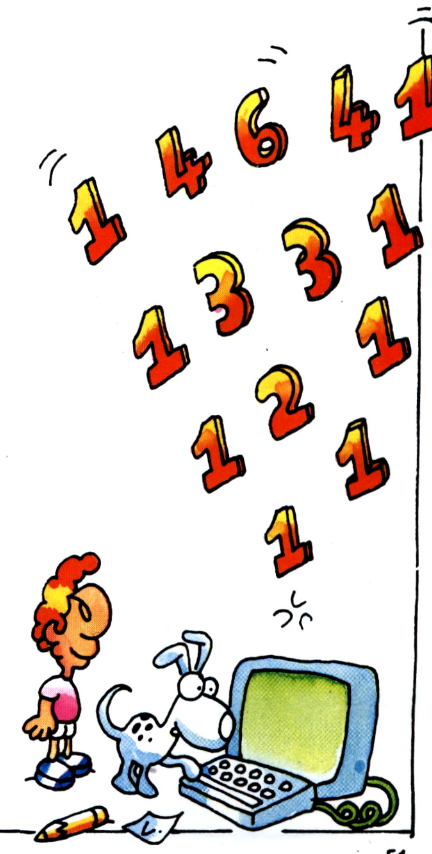
PRINT (= imprime). Indica al ordenador que «escriba» en la pantalla palabras, números, resultados de operaciones y de ecuaciones. Después de esta instrucción el ordenador imprime lo que le mandes. PRINT puede visualizar también el contenido de variables numéricas (A, B, etc.) y de variables de caracteres (A$, B$, etc.).

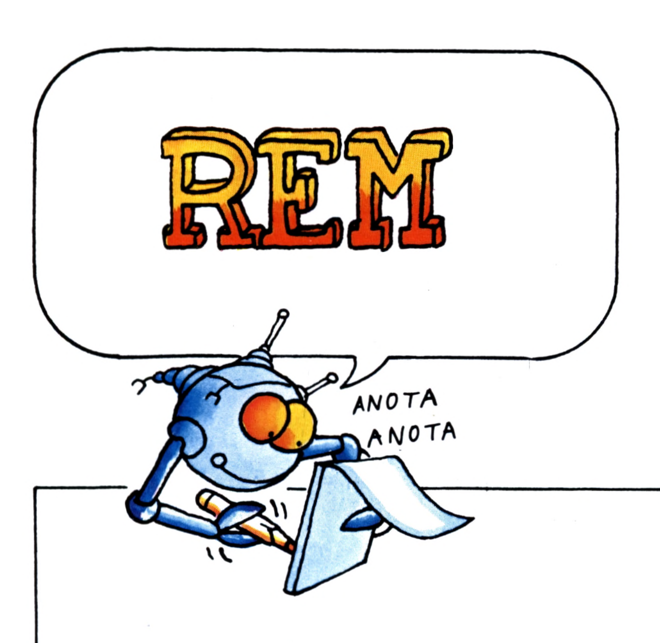


UN TRIANGULO CONTROVERTIDO

Unos dicen que ya lo conocían los árabes. Otros atribuyen el descubrimiento al italiano Tartaglia y otros al francés Pascal. Es un triángulo formado por números con propiedades singulares. Con este programita podréis crearlo y estudiarlo con el ordenador.

```
10 REM EL TRIANGULO DE TARTAGLIA
20 PRINT"CUANTAS LINEAS QUIERES?"
30 INPUT N
40 DIM P(N,N)
50 P(1,1)=1
60 REM IMPRIME EL PRIMER ELEMENTO
70 PRINT:PRINT 1
80 FOR K=2 TO N
90 P(K,1)=1
100 REM LA PRIMERA COLUMNA = 1
110 PRINT P(K,1);
120 FOR J=2 TO K
130 REM PREPARO EL RESTO DE LOS ELEMENTOS
140 P(K,J)=P(K-1,J)+P(K-1,J-1)
150 PRINT P(K,J);
160 NEXT J
170 PRINT
180 NEXT K
190 END
```





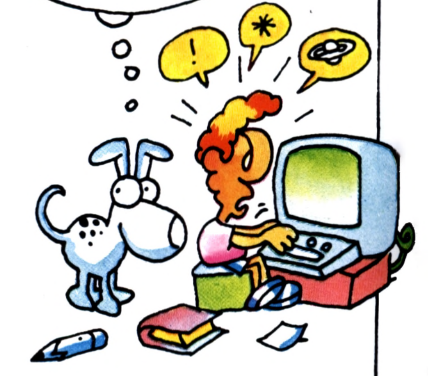
REM (de REMark = anotar). Después de REM se pueden introducir comentarios para quien escribe o lee el programa. Cuando el programa está partido, si el ordenador encuentra REM, no hace nada y pasa a la línea siguiente.



LOS NUMEROS PERFECTOS

Se llama «número perfecto» al número que es igual a la suma de todos sus divisores. Por ejemplo, 6 es un número perfecto, porque sus divisores, 3, 2 y 1 suman 6. He aquí un programa que busca los números perfectos entre 1 y N.

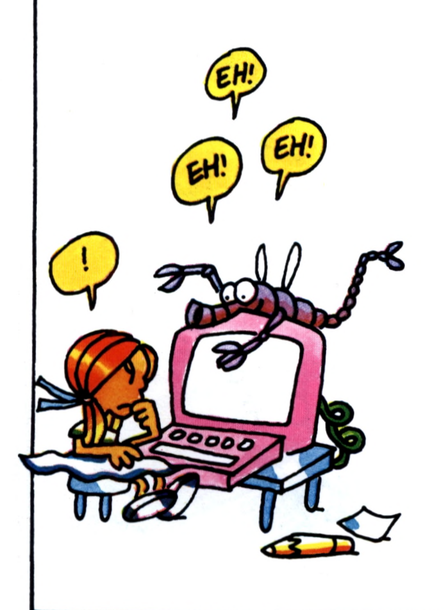
```
10 REM LOS NUMEROS PERFECTOS
20 REM 1 ES EL MENOR DE LOS NUMEROS PERFECTOS
30 CLS:PRINT"INTRODUCE EL NUMERO HASTA EL QUE HAY QUE BUSCAR"
40 INPUT N
50 FOR K=1 TO N
60 REM PONGO S=1
70 S=1
80 REM CALCULO K/2
90 D=INT(K/2)
100 FOR J=2 TO D
110 REM BUSCO LOS DIVISORES DE K
120 IF K/J<>INT(K/J) GOTO 150
130 REM SI J ES UN DIVISOR, LO SUMO
140 S=S+J
150 NEXT J
160 REM SI K<>S ENTONCES NO ES PERFECTO
170 IF K<>S GOTO 210
180 REM IMPRIMO EL NUMERO K
190 PRINT"NUMERO PERFECTO :";K
200 REM IMPRIMO UN MENSAJE
210 IF K/10=INT(K/10) THEN PRINT"ESTOY CALCULANDO"
220 REM CONSIDERO EL NUMERO SIGUIENTE
230 NEXT K
240 REM EL PROGRAMA SE ACABO
250 END
```



RETURN (= regreso). Es una instrucción BASIC que se usa sólo para que el ordenador vuelva de un subprograma al programa principal. En este libro lo hemos usado en el juego del NIM (ved GO SUB).





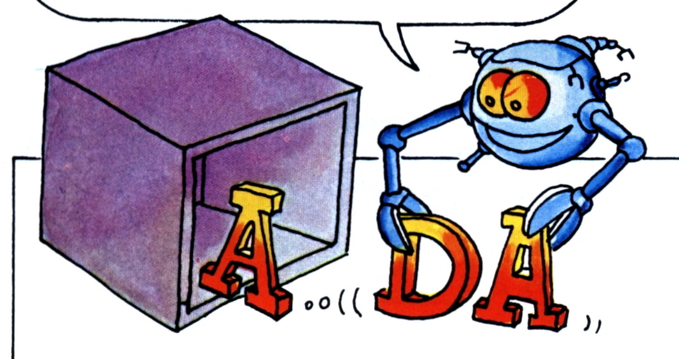


¡OJO CON LA TECLA RETURN!

La tecla RETURN equivale a algo así como a empezar desde el principio en las máquinas de escribir. En algunos ordenadores, en lugar de RETURN existe la tecla ENTER, ENTREE o simplemente una flecha. Pero es una tecla importantísima en todos los ordenadores. Sólo cuando hayáis pulsado RETURN (o la tecla equivalente), el ordenador realizará la orden que le habéis dado.
Es indispensable también mientras escribís un programa. Cada vez que escribís una línea, acordaos de pulsar siempre RETURN para empezar de nuevo. De lo contrario, cuando volváis para atrás el programa, el ordenador os indicará que hay un error.



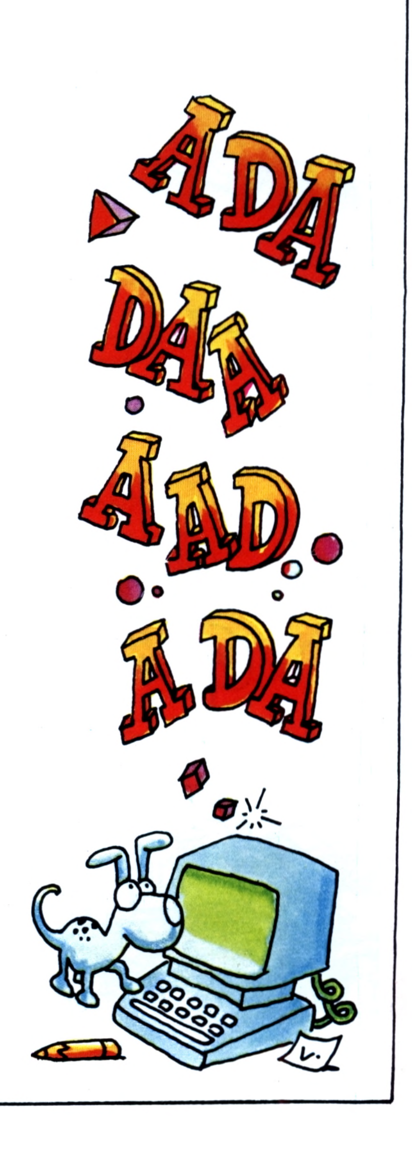
RIGHT$ (= derecha). Indica al ordenador que saque la parte más a la derecha de una cadena de caracteres. Si P$ = "ADA", RIGHT$ (P$,2) es "DA". En el Spectrum RIGHT$ se omite y deberemos escribir P$ (2 TO 3). En el ejemplo de abajo, con RIGHT$ se saca la parte más a la derecha de la palabra para obtener el anagrama.



LOS ANAGRAMAS

Sacar el anagrama quiere decir intercambiar las letras de todas las formas posibles para hacer otras palabras. Con este programita, limitado a las palabras de tres letras, el ordenador lo hará automáticamente.

```
10 REM ANAGRAMAS DE UNA PALABRA DE TRES LETRAS
20 DIM A$(3)
30 CLS:PRINT"INTRODUCE UNA PALABRA DE TRES LETRAS"
40 INPUT P$
50 REM METO LAS LETRAS EN LOS ELEMENTOS DE A$
60 PRINT
70 FOR K=1 TO 3
80 A$(K)=LEFT$(P$,1)
90 P$=RIGHT$(P$,3-K)
100 NEXT K
110 REM ANAGRAMA
120 FOR K=1 TO 3
130 FOR I=1 TO 3
140 IF I=K GOTO 170
150 J=6-(K+I)
160 PRINT A$(K);A$(I);A$(J)
170 NEXT I
180 NEXT K
190 END
```



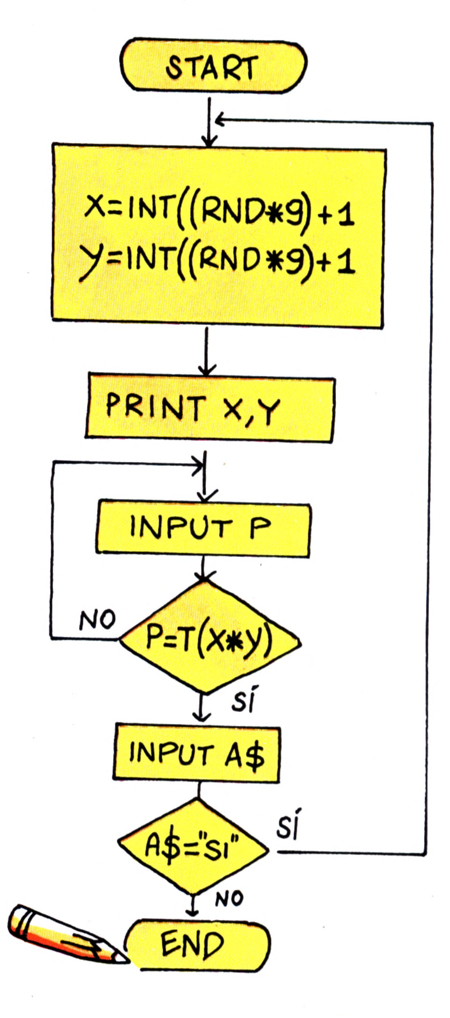
RND (de random = aleatorio). Indica al ordenador que genere un número cualquiera. El número que salga tiene que estar comprendido entre 0 y 1. Por este motivo, para conseguir números aleatorios enteros hay que usar también la función INT, y multiplicar RND por valores enteros.

¿QUEREIS UN ORDENADOR-PROFESOR?

Con este programita el ordenador se convierte en un profesor de matemáticas, que os preguntará la tabla del 1 al 9. Probadlo, aunque sepáis bien la tabla.

```
10 PRINT"APRENDEMOS LAS TABLAS"
20 CLS
30 X=INT(RND*(9+1))
40 Y=INT(RND*(9+1))
50 PRINT"DIME EL PRODUCTO DE LOS NUMEROS :"
60 PRINT:PRINT X,Y
70 PRINT:INPUT P
80 IF P=X*Y GOTO 110
90 PRINT:PRINT"RESPUESTA EQUIVOCADA. PRUEBA OTRA VEZ"
100 GOTO 70
110 PRINT:PRINT"RESPUESTA EXACTA."
115 PRINT:"QUIERES SEGUIR? SI/NO"
120 INPUT A$
130 IF A$="SI" GOTO 20
140 END
```





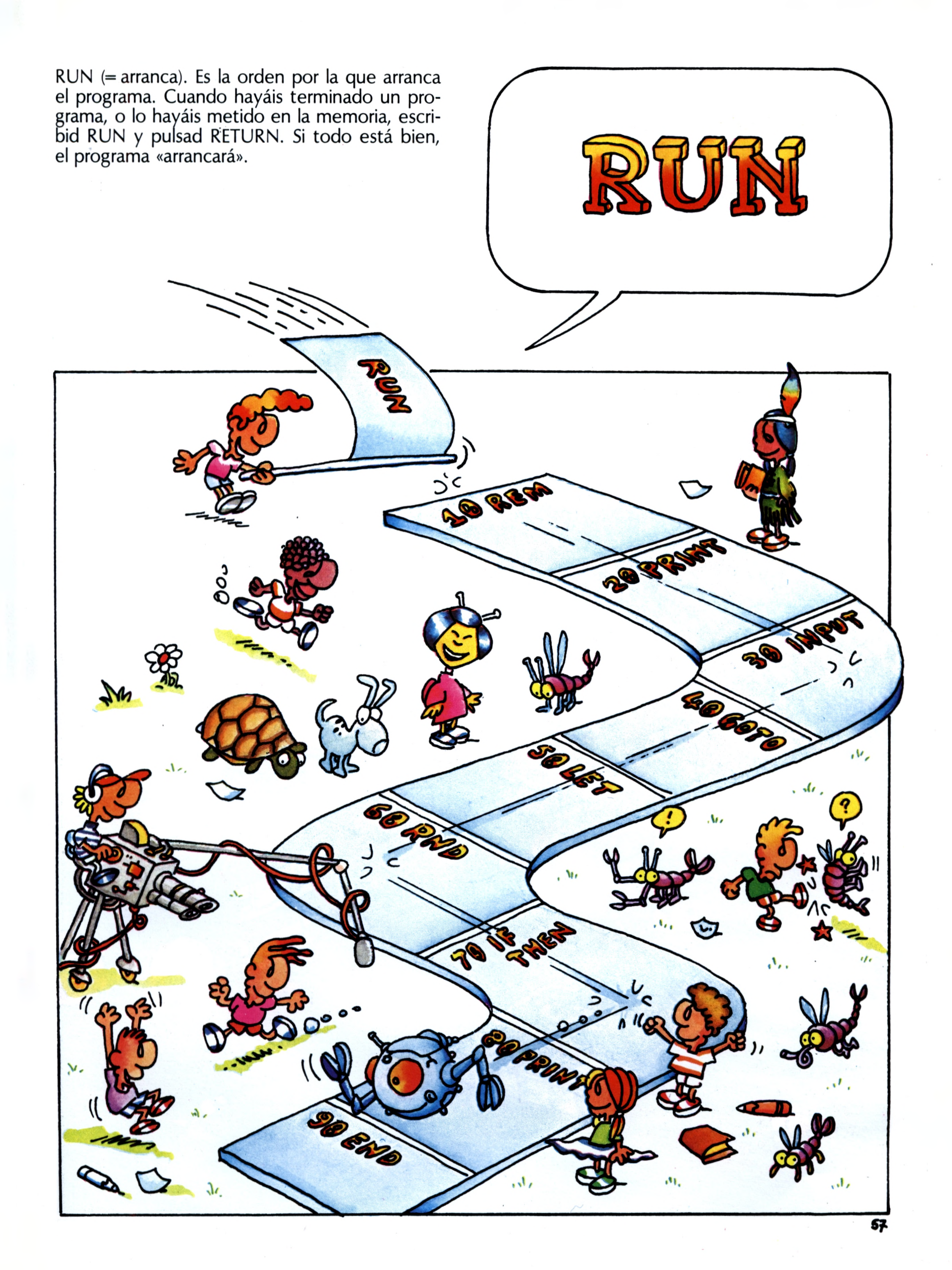


RANDOMIZE (o RANDOM)

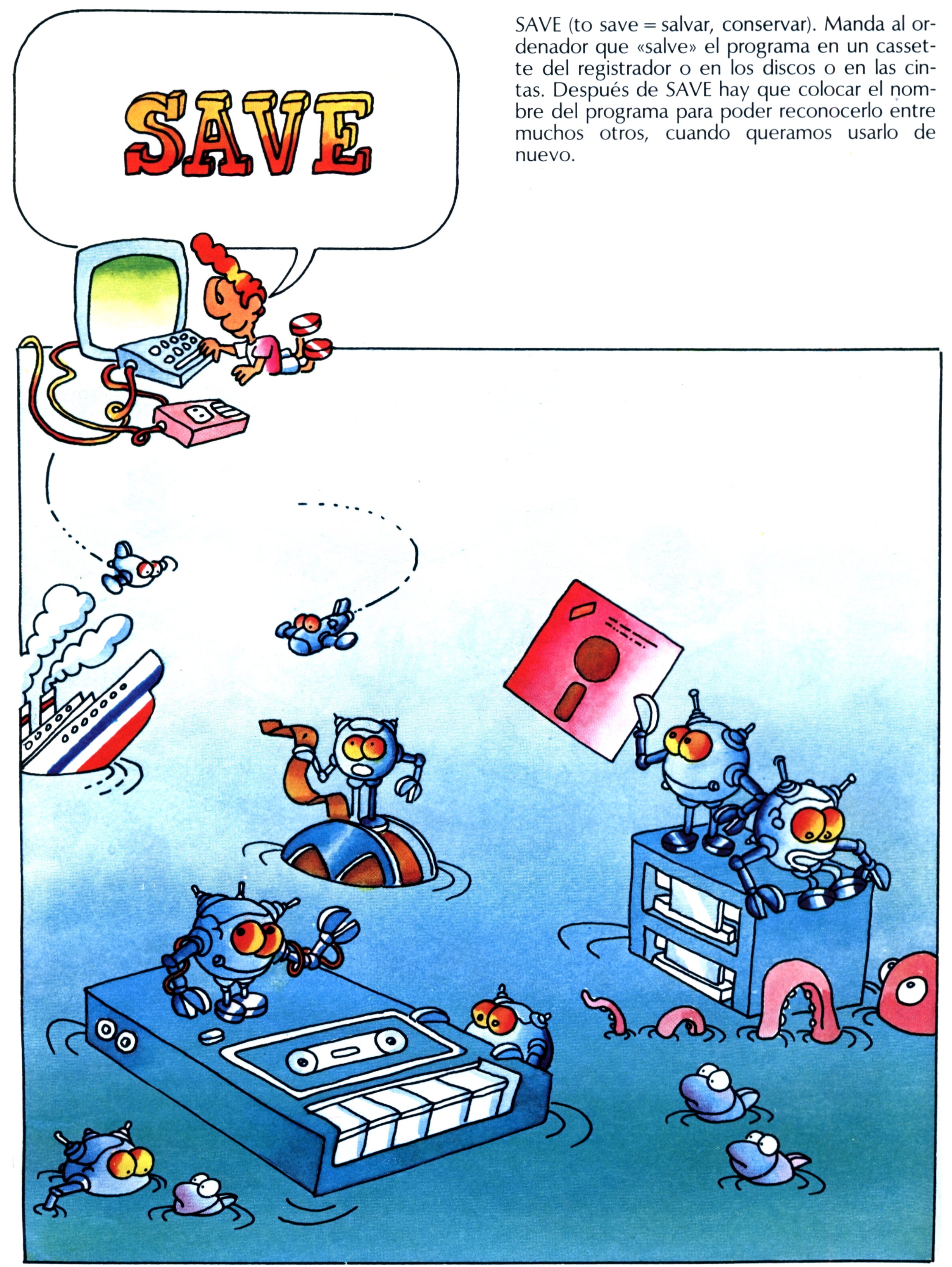
Pocos ordenadores dan los números aleatorios sólo con RND. Pues, efectivamente, en cada vuelta del programa pescan de sus celdillas de memoria siempre las mismas secuencias de números. Para conseguir secuencias realmente distintas hay que anteponer RANDOMIZE o RANDOM, depende del «dialecto» que hable vuestro ordenador.



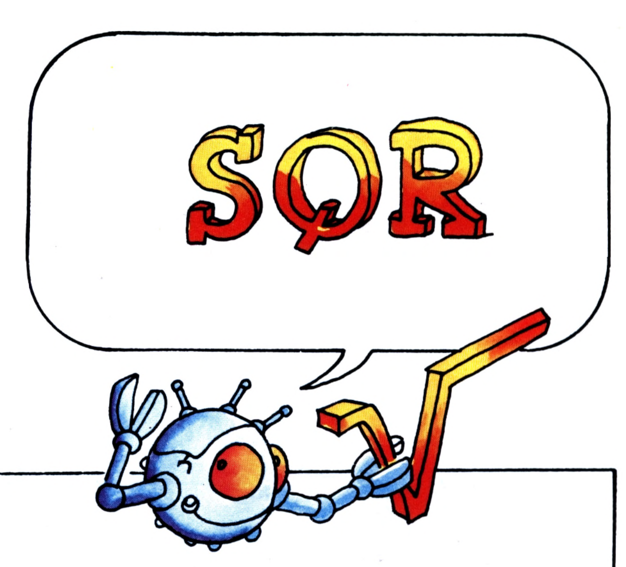
RUN (= arranca). Es la orden por la que arranca el programa. Cuando hayáis terminado un programa, o lo hayáis metido en la memoria, escribid RUN y pulsad RETURN. Si todo está bien, el programa «arrancará».



SAVE (to save = salvar, conservar). Manda al ordenador que «salve» el programa en un cassette del registrador o en los discos o en las cintas. Después de SAVE hay que colocar el nombre del programa para poder reconocerlo entre muchos otros, cuando queramos usarlo de nuevo.

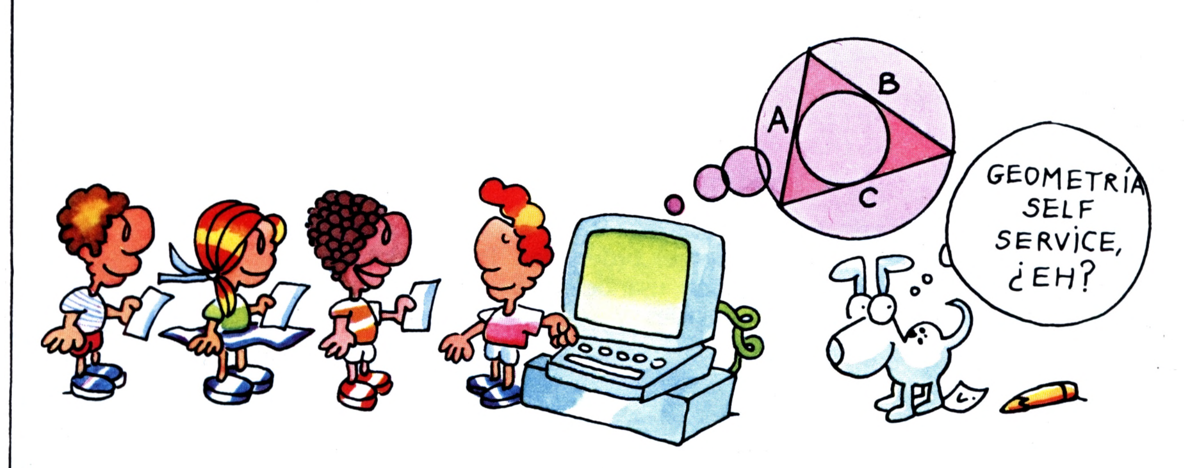


SQR. Calcula la raíz cuadrada de los números positivos. Por ejemplo, SQR(4) es 2.



PROBLEMITA

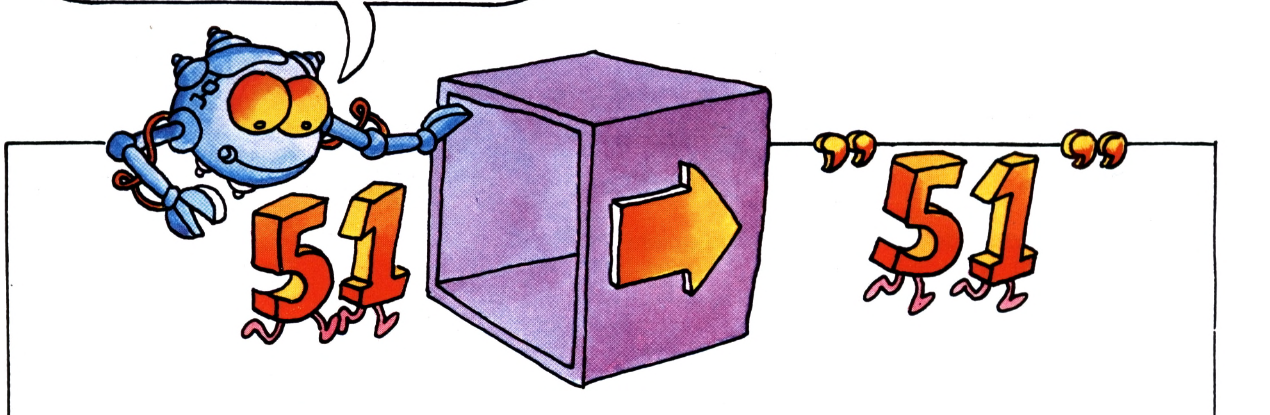
Dados los lados de un triángulo cualquiera ABC, hallar el área del círculo inscrito y del círculo circunscrito. He aquí el programa para resolverlo de una vez para siempre.




```
10 REM PROPIEDADES GEOMETRICAS DE UN TRIANGULO
20 CLS:PRINT "INTRODUCE LAS MEDIDAS DE LOS LADOS"
30 INPUT A,B,C
40 P=3.14159
50 S=(A+B+C)/2
60 A1=SOR(S*(S-A)*(S-B)*(S-C))
70 PRINT:PRINT "AREA DEL TRIANGULO = ";A1
80 R1=A1/S
90 A2=P*R1^2
100 PRINT:PRINT "AREA DEL CIRCULO MAXIMO INSCRITO = ";A2
110 R2=A*B*C/(4*A1)
120 A3=P*R2^2
130 PRINT:PRINT"AREA DEL CIRCULO MINIMO CIRCUNSCRITO = ";A3
140 END
```

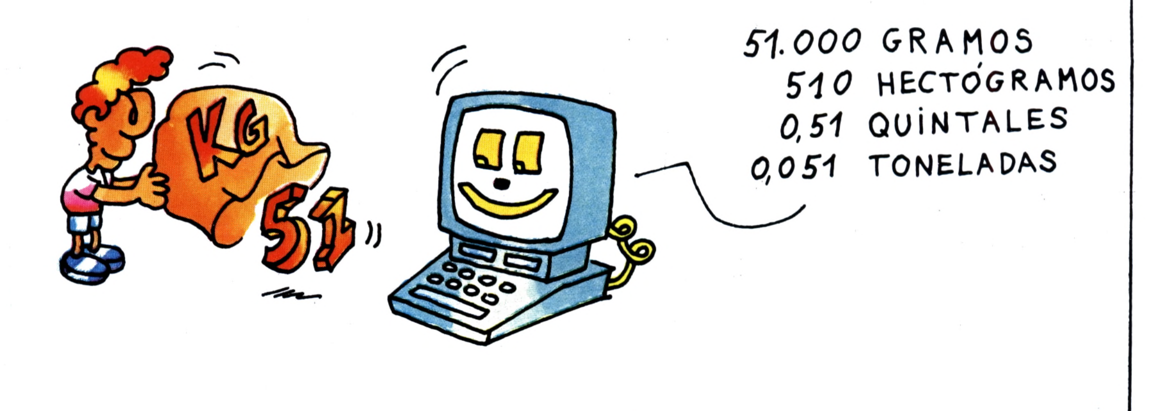


STR$ (de STRing = cadena). Indica al ordenador que trate un valor numérico como cadena. STR$ (51) es "51".



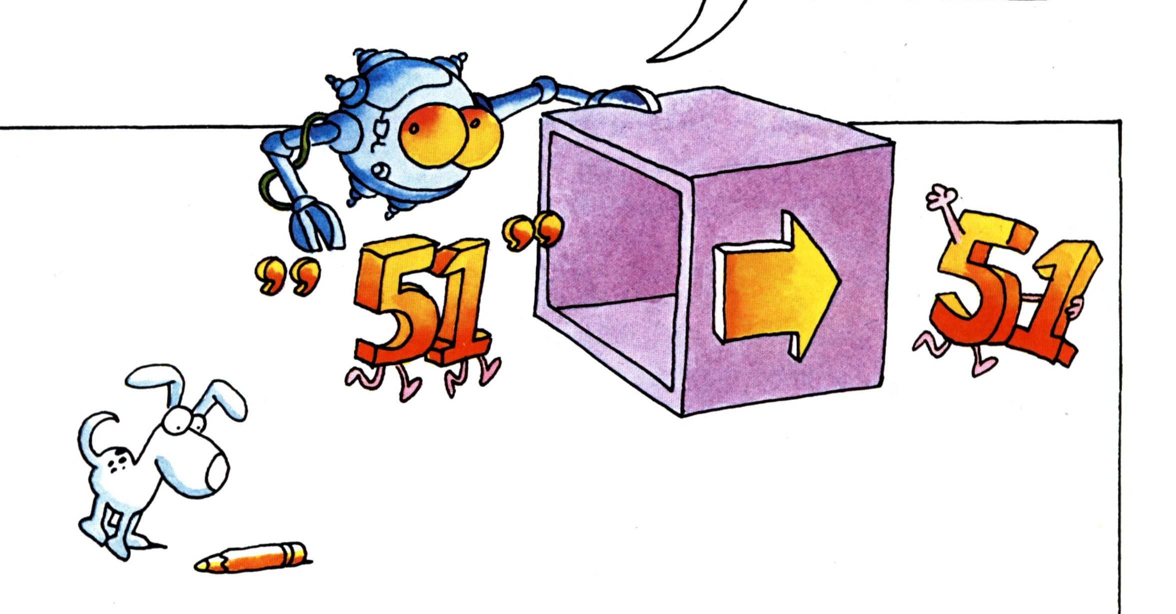
LAS EQUIVALENCIAS

¿Cuántas veces en el colegio tuvisteis que hacer las equivalencias? Ahora con este programita os las puede hacer el ordenador. Podréis introducir cualquier medida de peso y el ordenador, usando STR$ y VAL, sacará de las cadenas los valores numéricos esenciales y realizará las equivalencias por vosotros.



VAL (de value = valor). Indica al ordenador que trate como valores numéricos los números escritos como cadenas. Por ejemplo, si A$ = "51", VAL (A$) es el número 51. VAL es la función inversa de STR$.





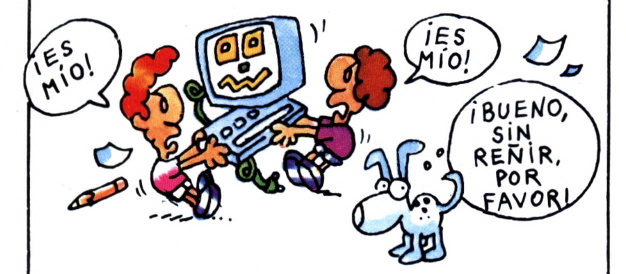

```
10 REM LAS EQUIVALENCIAS
20 CLS
30 PRINT:PRINT"INTRODUCE UNA MEDIDA DE PESO"
40 INPUT A$
50 N=VAL(A$)
60 N$=STR$(N)
70 L=LEN(N$)
80 M=LEN(A$)
90 A$=RIGHT$(A$,M-L)
100 A$=LEFT$(A$,3)
110 IF A$="GRA" THEN N=N*1:GOTO 170
120 IF A$="HEC " THEN N=N*100:GOTO 170
130 IF A$="KIL" THEN N=N*1000:GOTO 170
140 IF A$="QUI" THEN N=N*100000!:GOTO 170
150 IF A$="TON" THEN N=N*1E+06:GOTO 170
160 PRINT:PRINT"NO CONOZCO ESTA UNIDAD DE MEDIDA":GOTO 30
170 PRINT:PRINT"EN QUE UNIDADES QUIERES QUE SE EXPRESE EL RESULTADO?"
180 INPUT B$
190 B$=LEFT$(B$,3)
200 IF B$="GRA" THEN N=N/1:C$="GRAMOS":GOTO 260
210 IF B$="HEC" THEN N=N/100:C$="HECTOGRAMOS":GOTO 260
220 IF B$="KIL" THEN N=N/1000:C$="KILOGRAMOS":GOTO 260
230 IF B$="QUI" THEN N=N/100000!:C$="QUINTALES":GOTO 260
240 IF B$="TON" THEN N=N/1E+06:C$="TONELADAS":GOTO 260
250 PRINT:PRINT"NO CONOZCO ESTA UNIDAD DE MEDIDA":GOTO 170
260 PRINT:PRINT"EL RESULTADO EXPRESADO EN ";C$;" ES:"; N
270 PRINT:PRINT:PRINT"DESEAS OTRA EQUIVALENCIA?"
280 INPUT R$
290 IF R$="SI" GOTO 20
```

COMO SE USA ESTE LIBRO

1. EN PRIMER LUGAR SE HOJEA Y SE LEE. DESCUBRIRÉIS QUE EL BASIC NO ES UNA LENGUA TAN DIFÍCIL.

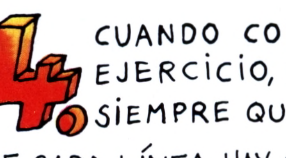


2. LUEGO CONSEGUID UN ORDENADOR, CUALQUIER ORDENADOR, EL VUESTRO, EL DE PAPÁ, EL DEL COLEGIO, EL DEL AMIGO...



3. RESUELTO EL PROBLEMA DEL ORDENADOR, COPIAD LOS "PROGRAMITAS" DEL LIBRO EMPEZANDO POR LOS MÁS BONITOS.





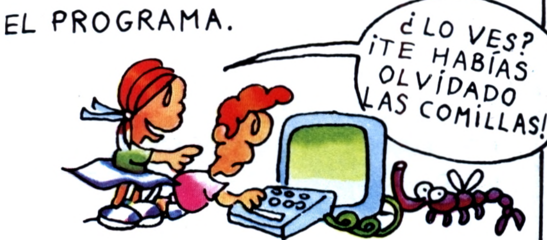
4. CUANDO COPIÁIS UN EJERCICIO, RECORDAD SIEMPRE QUE AL FINAL DE CADA LÍNEA HAY QUE PULSAR **RETURN** (PÁG. 53).

EN CAMBIO, PARA QUE "ARRANQUE" EL PROGRAMA, TENÉIS QUE ESCRIBIR **RUN** (PÁG. 57) Y LUEGO RETURN.

5.



SI APARECE UN MENSAJE DE ERROR, ESCRIBID **LIST** (PÁG. 40) PARA RECLAMAR LA LÍNEA EQUIVOCADA O TODO EL PROGRAMA.



6. CORREGID LAS LÍNEAS EQUIVOCADAS, ESCRIBID DE NUEVO RUN, PULSAD RETURN Y... POR FIN EL PROGRAMA MARCHARÁ.



7. EN ESTE LIBRO LAS "PALABRAS" DE BASIC ESTÁN COLOCADAS EN ORDEN ALFABÉTICO PARA FORMAR UN BUEN DICCIONARIO.

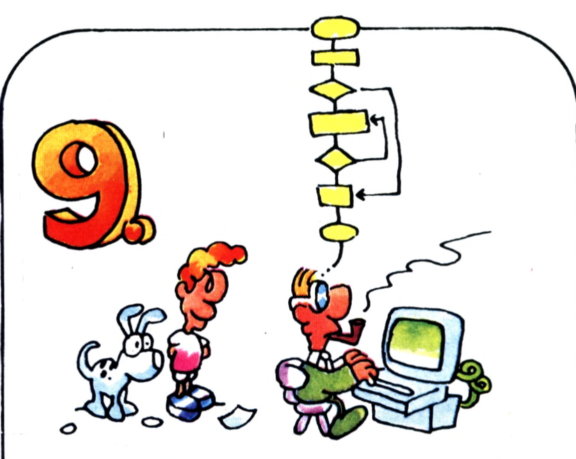
DE ESTA FORMA, CUANDO ENCONTRÉIS UNA INSTRUCCIÓN O UN VOCABLO QUE NO CONOCÉIS, PODÉIS HALLARLO FÁCILMENTE EN EL "DICCIONARIO" O EN EL ÍNDICE ANALÍTICO (PÁG. 64).



8. SI ALGÚN PROGRAMA INSISTE EN SER PROBLEMA, AYUDAOS CON EL MANUAL DE VUESTRO ORDENADOR. PUEDE HABER PEQUEÑAS DIFERENCIAS ENTRE EL "DIALECTO BASIC" DE VUESTRO ORDENADOR Y EL DE NUESTROS PROGRAMAS. UNA COMPARACIÓN SERÁ ÚTIL.



9.



ALGUNOS PROGRAMAS VAN ACOMPAÑADOS DE SUS "DIAGRAMAS DE FLUJO." ESTOS ESQUEMAS LLEVAN SÓLO LOS PASAJES LÓGICOS FUNDAMENTALES, LOS "PRINT" Y LOS "REM" NO ESTÁN REPRESENTADOS. CONFRONTÁNDOLOS CON LOS PROGRAMAS, APRENDERÉIS A PROGRAMAR.

10. HAY "EJERCICIOS" PARA TODOS LOS GUSTOS. ALGUNOS SON "ESCOLARES", OTROS SON JUEGOS DIVERTIDOS. PODÉIS "SALVAR" LOS QUE MÁS OS GUSTEN CON **SAVE** (PÁG. 58) Y COMENZAR ASÍ VUESTRA BIBLIOTECA DE PROGRAMAS.

EN RESUMEN, SI EMPLEÁIS BIEN ESTE LIBRO, AL FINAL SABRÉIS BASTANTE BASIC Y OS ENVIDIARÁ PAPÁ.



INDICE ANALITICO

En este libro, Ada y Marko, en compañía del perrito Plotter y de un ordenador personal, alegre y atento, emprenden el viaje coloreado y divertido a través de la informática que habían empezado en **Mi primer libro sobre ordenadores,** del mismo autor, Luca Novelli.

El BASIC es un lenguaje inventado por los informáticos para indicar al or[illegible]or lo que debe hacer; en otras palabras, para escribir «progra-[illegible] autor lo ha transformado en un gran juego, fascinante y diver-[illegible] programa normal BASIC se convierte de esta forma en un va-[illegible]nto [illegible]juego de la oca», en el que se pueden recorrer itinerarios lentos o rápidos, atajos astutos, idas y vueltas extenuantes y pasos rápidos, envueltos en insidiosas trampas (los famosos «bug» o errores de programación).

El libro es un buen diccionario de BASIC, y para cada vocablo hay un programa o un juego. Es una invitación a los jóvenes y a los más pequeños para que sean creativos con el ordenador en casa y en el colegio. Y es también una invitación a todos los que deseen introducirse en el mundo de la programación y de los «lenguajes» para hablar con el ordenador. Es un libro que se puede leer de un tirón, pero en particular es un amigo que se debe tener cerca del ordenador para «aprender haciendo».

Luca Novelli, popular autor italiano de historietas ilustradas y de tiras de comics, desde 1977 escribe e ilustra libros de divulgación científica para niños.


Mi primer libro de BASIC
LUCA NOVELLI